



# UNIVERSIDAD DE CUENCA

Facultad de Ingeniería  
Carrera de Ingeniería de Sistemas

## Creación de una arquitectura enfocada en la interacción hombre-computador para aplicaciones de internet de las cosas en ambientes de vida asistidos

Trabajo de titulación previo a la obtención del título de  
Ingeniero de Sistemas

### AUTORES:

Fernando Xavier Garnica Bautista  
C.I. 0105890404

Edwin Xavier Maita Tepán  
C.I. 0105661326

### DIRECTORA:

Ing. Irene Priscila Cedillo Orellana, PhD.  
C.I. 0102815842

CUENCA - ECUADOR  
2018



# Resumen

Debido a que la Interacción Hombre Computador (HCI) representa un canal de comunicación directo entre las personas y las aplicaciones tecnológicas, es necesario que la tecnología tenga un enfoque centrado en brindar soluciones que faciliten la realización de actividades cotidianas y que integren a individuos de diferentes sectores de la población. En este sentido, se han creado paradigmas como Vida Asistida por el Entorno (AAL) e Internet de las Cosas (IoT) que, usados en conjunto, buscan brindar un soporte hacia una mejor calidad de vida de las personas de sectores vulnerables como los adultos mayores y las personas con discapacidad, mediante tecnologías que interactúen con las personas de forma natural y transparente en el lugar y contexto donde el usuario las necesite, ayudándoles a llevar una vida independiente y satisfactoria.

En este trabajo de titulación, se propone una arquitectura de software que sirve de soporte durante la construcción de soluciones que faciliten la interacción entre usuarios con necesidades especiales y diferentes tipos de dispositivos, sensores o actuadores conectados a Internet dentro de entornos de vida asistida. Esta arquitectura contempla diferentes modos de interacción (gestual, auditiva, visual, cerebral, entre otras) y las interconecta con dispositivos que permiten a las personas realizar acciones cotidianas que, en otras circunstancias, pueden resultar difíciles para ellas.

Para la evaluación de la arquitectura propuesta, se ha realizado la implementación del prototipo de un sistema que se basa en la solución presentada en este trabajo de titulación y que además ha permitido la ejecución de un cuasi-experimento que permite analizar la arquitectura desde el punto de vista de las percepciones del usuario.

**Palabras Clave:** Arquitectura de software, Interacción hombre-computadora, Internet de las cosas, Vida Asistida por el Entorno.



# Abstract

Because of the Interaction Human Computer (HCI) represents a direct communication channel between people and technological applications, it is necessary for technology to have a focus on providing solutions that facilitate the realization of daily activities, not only to ordinary users, but also to people who cannot interact with different applications in the same way that a person who has all their physical and cognitive faculties does. In this sense, paradigms have been created such as Ambient Assisted Living (AAL) and Internet of Things (IoT) that, used together, seek to promote and prolong the independent life of people from vulnerable sectors such as the elderly people and people with disabilities, through technologies that interact with people in a natural and transparent way in the place and context where the user needs them.

In this work, a software architecture is proposed that serves as support during the construction of solutions that facilitate the interaction between users with special needs and different types of devices, sensors or actuators connected to the Internet within assisted living environments, in such a way that they contribute to improve their quality of life.

For the validation, a prototype implementation of a system based on the proposed architecture has been carried out, giving way to a quasi-experiment that allows analyzing the architecture from the point of view of the user's perceptions.

**Keywords:** Software architecture, Human-computer interaction, Internet of things, Ambient Assisted Living.



# Índice general

<b>Índice de Figuras</b>	<b>VII</b>
<b>Índice de Tablas</b>	<b>XI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación y contexto . . . . .	2
1.2. Planteamiento del problema . . . . .	4
1.3. Solución propuesta . . . . .	5
1.4. Hipótesis y objetivos . . . . .	5
1.4.1. Hipótesis . . . . .	5
1.4.2. Objetivos . . . . .	5
1.5. Metodología de la investigación . . . . .	6
1.6. Estructura del trabajo . . . . .	7
1.7. Aporte científico . . . . .	9
<b>2. Marco tecnológico</b>	<b>10</b>
2.1. Vida asistida por el entorno . . . . .	11
2.1.1. Inteligencia ambiental . . . . .	11
2.1.2. Objetivos de Vida Asistida por el Entorno . . . . .	12
2.2. Internet de las cosas . . . . .	12
2.2.1. Características . . . . .	13
2.2.2. Modelo de referencia . . . . .	13
2.2.3. Dispositivos . . . . .	15
2.3. Interacción humano-computadora . . . . .	16
2.3.1. Fundamentos . . . . .	17
2.3.2. Paradigmas y estilos de interacción . . . . .	17
2.3.3. Interacción natural . . . . .	18
2.4. e-Health . . . . .	23
2.5. Arquitectura de software . . . . .	24





2.5.1. Componente . . . . .	24
2.5.2. Módulo . . . . .	25
<b>3. Estado del arte</b>	<b>26</b>
3.1. Metodología . . . . .	27
3.2. Etapa de planificación . . . . .	27
3.2.1. Preguntas de investigación . . . . .	27
3.2.2. Estrategia de búsqueda . . . . .	28
3.2.3. Cadena de búsqueda . . . . .	28
3.2.4. Período de búsqueda . . . . .	29
3.2.5. Criterios de extracción de datos . . . . .	29
3.3. Etapa de ejecución . . . . .	32
3.3.1. Selección de estudios primarios . . . . .	32
3.3.2. Aseguramiento de la calidad . . . . .	33
3.4. Etapa de reporte . . . . .	33
3.4.1. Porcentajes individuales . . . . .	34
3.4.2. Comparación de criterios . . . . .	41
3.4.3. Discusión . . . . .	44
3.5. Conclusiones . . . . .	47
<b>4. Arquitectura de software: AALIIoT</b>	<b>49</b>
4.1. Descripción general . . . . .	50
4.2. Vista estática . . . . .	52
4.2.1. Módulo Gestor de Usuarios . . . . .	52
4.2.2. Módulo Receptor de Interacción . . . . .	53
4.2.3. Middleware de Interacción . . . . .	54
4.2.4. Módulo Controlador de Ejecución . . . . .	55
4.3. Vista funcional . . . . .	56
4.3.1. Componentes del Módulo Gestor de Usuarios . . . . .	56
4.3.2. Componentes del Módulo Receptor de Interacción . . . . .	63
4.3.3. Componentes del Middleware de Interacción . . . . .	65
4.3.4. Componentes del Módulo Controlador de Ejecución . . . . .	68
4.3.5. Descripción de los controladores de interacción . . . . .	71
4.3.6. Protocolos de comunicación . . . . .	75
4.3.7. Diagrama de componentes . . . . .	76
4.4. Vista dinámica . . . . .	77
4.4.1. Diagrama de actividad del Módulo Gestor de Usuarios . . . . .	78
4.4.2. Diagrama de actividad del Módulo Receptor de Interacción . . . . .	78
4.4.3. Diagrama de actividad del Middleware de Interacción . . . . .	80
4.4.4. Diagrama de actividad del Módulo Controlador de Ejecución . . . . .	82



<b>5. Implementación</b>	<b>83</b>
5.1. Escenario . . . . .	84
5.2. Perspectiva del sistema . . . . .	84
5.3. Descripción de la funcionalidad del sistema . . . . .	84
5.3.1. Requerimientos funcionales de usuario (RF) . . . . .	84
5.3.2. Casos de uso . . . . .	85
5.3.3. Diagramas de secuencia . . . . .	86
5.4. Implementación del prototipo del sistema . . . . .	91
5.4.1. Módulo receptor de interacción . . . . .	93
5.4.2. Middleware de interacción . . . . .	97
5.4.3. Módulo Controlador de ejecución . . . . .	102
5.5. Despliegue del prototipo del sistema . . . . .	107
5.6. Conclusiones . . . . .	109
<b>6. Evaluación empírica</b>	<b>110</b>
6.1. Introducción . . . . .	111
6.2. Modelos teóricos de evaluación en Ingeniería de <i>Software</i> . . . .	112
6.2.1. <i>Technological acceptance model</i> (TAM) . . . . .	112
6.2.2. <i>Method Evaluation Model</i> (MEM) . . . . .	114
6.3. Adaptando el MEM para su uso en arquitecturas de <i>Software</i> .	115
6.4. Evaluando la utilidad percibida de la arquitectura de software AALIIoT en la práctica mediante cuasi-experimentos . . . . .	120
6.4.1. Planificación del cuasi-experimento . . . . .	122
6.4.2. Operación y ejecución del cuasi-experimento . . . . .	125
6.4.3. Ejecución y análisis de los cuasi-experimentos individuales	126
6.4.4. Análisis de los resultados . . . . .	138
6.5. Amenazas a la validez . . . . .	145
6.5.1. Validez interna . . . . .	145
6.5.2. Validez externa . . . . .	146
6.5.3. Validez del constructo . . . . .	146
6.5.4. Validez de la conclusión . . . . .	147
6.6. Conclusiones . . . . .	147
<b>7. Conclusiones y trabajos futuros</b>	<b>149</b>
7.1. Conclusiones . . . . .	150
7.1.1. Objetivo general . . . . .	150
7.1.2. Objetivos específicos . . . . .	151
7.1.3. Hipótesis . . . . .	154
7.2. Trabajos futuros . . . . .	155
7.2.1. Respecto a la arquitectura de software AALIIoT . . . .	155
7.2.2. Respecto al prototipo del sistema implementado . . . .	155



7.2.3. Respecto a la validación y mejora en las evaluaciones empíricas . . . . .	156
<b>A. Estudios primarios seleccionados</b>	<b>157</b>
A.1. IEEE Xplore . . . . .	157
A.2. ACM Digital Library . . . . .	167
A.3. Science Direct . . . . .	169
A.4. Springer Link . . . . .	170
<b>B. Porcentajes de los criterios de extracción</b>	<b>182</b>
<b>C. Diagramas UML</b>	<b>188</b>
C.1. Diagrama de componentes . . . . .	188
C.2. Diagrama de actividades . . . . .	189
C.3. Diagrama de caso de uso . . . . .	191
C.4. Diagrama de secuencia . . . . .	192
<b>D. Herramientas</b>	<b>195</b>
D.1. AppInventor . . . . .	195
D.2. Módulo Wifi ESP8266 . . . . .	195
D.2.1. Características . . . . .	196
D.3. Módulo Relé . . . . .	196
D.4. IDE Arduino . . . . .	197
<b>E. Experimento Empírico</b>	<b>198</b>
E.1. Boletín de entrenamiento . . . . .	199
E.2. Cuestionario . . . . .	213
E.3. Resultados primer cuasi-experimento . . . . .	217
E.4. Resultados segundo cuasi-experimento . . . . .	218
<b>Bibliografía</b>	<b>220</b>



# Índice de Figuras

1.1. Personas con discapacidad en Ecuador según el tipo de discapacidad. Fuente: [22]. . . . .	3
1.2. Metodología de la investigación. Fuente: Elaboración propia. . .	7
1.3. Estructura del trabajo de titulación. Fuente: Elaboración propia.	9
2.1. Modelo de referencia. Fuente: [40]. . . . .	14
2.2. Interacción basada en la voz. Fuente: [5]. . . . .	19
2.3. Monitoreo eyetracking. Fuente: [38]. . . . .	20
2.4. Control de un sistema robótico usando BCI. Fuente: [41]. . . .	21
2.5. Sistema controlado por gestos que se ejecuta en un televisor 3D. Fuente: [27]. . . . .	22
2.6. Sistema basado en sensores para monitoreo cardíaco. Fuente: [81].	23
3.1. EC1 - Porcentaje de entornos para despliegue de soluciones IoT. Fuente: Elaboración propia. . . . .	35
3.2. EC3 - Porcentaje de partes interesadas a los que van dirigidas las soluciones IoT. Fuente: Elaboración propia. . . . .	36
3.3. EC7 - Porcentaje de tipos de soluciones IoT propuestas. Fuente: Elaboración propia. . . . .	38
3.4. EC14 - Porcentaje de tipos de interacción propuestas para con- trolar soluciones IoT. Fuente: Elaboración propia. . . . .	40
3.5. Comparación de EC2: Sector Vulnerable y EC1: Entornos entre EC14: Interacción. Fuente: Elaboración propia. . . . .	41
3.6. Comparación de EC7: Sistemas y EC14: Interacción. Fuente: Elaboración propia. . . . .	42
3.7. Comparación de EC12: Tratamientos y EC4: Accesorios entre EC2: Sector Vulnerable. Fuente: Elaboración propia. . . . .	43



3.8. Comparación de EC15: Ambiente de creación y EC18: Tipo de validación entre EC16: Modelo de solución propuesta. Fuente: Elaboración propia. . . . .	44
3.9. Cantidad de estudios por año seleccionados para el mapeo sistemático. Fuente: Elaboración propia. . . . .	45
4.1. Arquitectura de software AALIIoT. Fuente: Elaboración propia. . . . .	51
4.2. Módulo Gestor de Usuarios. Fuente: Elaboración propia. . . . .	52
4.3. Módulo Receptor de Interacción. Fuente: Elaboración propia. . . . .	53
4.4. Middleware de Interacción. Fuente: Elaboración propia. . . . .	54
4.5. Módulo Controlador de Ejecución. Fuente: Elaboración propia. . . . .	55
4.6. Diagrama de clases que define la estructura de un perfil de usuario. Fuente: Elaboración propia. . . . .	60
4.7. Ejemplo de la definición de un perfil de usuario en formato XML. Fuente: Elaboración propia. . . . .	62
4.8. Receptores de señales gestuales. a) Kinect b) Brazaletes MYO c) Dispositivo Leap. Fuente: [51], [57], [48]. . . . .	63
4.9. Gestos relacionados con acciones. Fuente: [65]. . . . .	72
4.10. Arquitectura que describe el funcionamiento básico de BCI. Fuente: Elaboración propia. . . . .	73
4.11. Proceso que describe el funcionamiento del control de interacción con GUIs. Fuente: Elaboración propia. . . . .	74
4.12. Búsqueda del gesto realizado por el usuario en el repositorio de comandos gestuales. Fuente: Elaboración propia. . . . .	75
4.13. Diagrama de componentes de la arquitectura AALIIoT. Fuente: Elaboración propia. . . . .	77
4.14. Diagrama de actividad del Módulo Gestor de Usuarios. Fuente: Elaboración propia. . . . .	78
4.15. Diagrama de actividad del Módulo Receptor de Interacción. Fuente: Elaboración propia. . . . .	79
4.16. Diagrama de actividad para captar interacción e identificar usuarios. Fuente: Elaboración propia. . . . .	80
4.17. Diagrama de actividad del Middleware de Interacción. Fuente: Elaboración propia. . . . .	81
4.18. Diagrama de actividad del Módulo Controlador de Ejecución. Fuente: Elaboración propia. . . . .	82
5.1. Diagrama de casos de uso para interacción gestual. Fuente: Elaboración propia. . . . .	85
5.2. Diagrama de casos de uso para interacción con GUIs. Fuente: Elaboración propia. . . . .	86



5.3. Diagrama de secuencia: Dispositivo receptor de interacción. Fuente: Elaboración propia. . . . .	87
5.4. Diagrama de secuencia: Controlador de interacción. Fuente: Elaboración propia. . . . .	88
5.5. Diagrama de secuencia: Identificador de dispositivo objetivo. Fuente: Elaboración propia. . . . .	89
5.6. Diagrama de secuencia: Middleware de interacción. Fuente: Elaboración propia. . . . .	90
5.7. Diagrama de secuencia: Controlador de ejecución. Fuente: Elaboración propia. . . . .	91
5.8. Aplicación móvil que funciona como Receptor de Interacción GUI. a) GUI principal, b) GUI de configuración, c) GUI de control. Fuente: Elaboración propia. . . . .	94
5.9. Programación por bloques de la GUI de Configuración. Fuente: Elaboración propia. . . . .	95
5.10. Programación por bloques del botón “Encender” de la GUI de control. Fuente: Elaboración propia. . . . .	96
5.11. Estructura del Middleware de interacción. Fuente: Elaboración propia. . . . .	97
5.12. Interfaz gráfica de usuario del Middleware de interacción. Fuente: Elaboración propia. . . . .	98
5.13. Implementación básica del Controlador de interacción GUI. Fuente: Elaboración propia. . . . .	99
5.14. Método que envía la solicitud de ejecución al Controlador de ejecución. Fuente: Elaboración propia. . . . .	100
5.15. Representación del repositorio de señales gestuales. Fuente: Elaboración propia. . . . .	100
5.16. a) Binarización de la imagen. b) Dibujar contorno. c) Polígono convexo y vértices. Fuente: Elaboración propia. . . . .	101
5.17. Código que aplica la regla de cosenos para contar los dedos mostrados en la mano (defectos o vértices). Fuente: Elaboración propia. . . . .	102
5.18. Ejemplo de conexión entre los Dispositivos Objetivo y el Controlador de ejecución. Fuente: Elaboración propia. . . . .	103
5.19. Configuración del módulo NodeMCU. Fuente: Elaboración propia.	104
5.20. Funciones que manejan las solicitudes de ejecución. Fuente: Elaboración propia. . . . .	105
5.21. Conexión entre la bombilla de luz y el módulo NodeMCU. Fuente: Elaboración propia. . . . .	106
5.22. Representación del Registro de dispositivos objetivo dentro del escenario planteado. Fuente: Elaboración propia. . . . .	107



5.23. Usuario controlando la bombilla de luz mediante gestos. Fuente: Elaboración propia. . . . .	108
5.24. Usuario controlando la bombilla de luz mediante una GUI. Fuente: Elaboración propia. . . . .	108
6.1. <i>Technology Acceptance Model</i> (TAM) simplificado. Fuente: [24].	113
6.2. <i>Method Evaluation Model</i> – MEM. Fuente: [56]. . . . .	114
6.3. Distribución de preguntas del cuestionario. Fuente: [19]. . . . .	117
6.4. Modelo teórico para la evaluación de la arquitectura AALIIoT. Fuente: [19]. . . . .	118
6.5. Resumen de los cuasi-experimentos realizados. Fuente: Elaboración propia. . . . .	126
6.6. UC1 - Diagrama de cajas para las variables PEOU, PU e ITU. Fuente: Elaboración propia. . . . .	127
6.7. UC2 - Diagrama de cajas para las variables PEOU, PU e ITU. Fuente: Elaboración propia. . . . .	133
6.8. Diagrama de cajas para la variable Eficiencia en los cuasi-experimentos realizados. Fuente: Elaboración propia. . . . .	139
6.9. Diagrama de cajas para la variable Efectividad en los cuasi-experimentos realizados. Fuente: Elaboración propia. . . . .	140
6.10. UC1 - Conclusiones de la aplicación de MEM a la Arquitectura propuesta. Fuente: Elaboración propia. . . . .	144
6.11. UC2 - Conclusiones de la aplicación de MEM a la Arquitectura propuesta. Fuente: Elaboración propia. . . . .	145
C.1. Ejemplo de diagrama de componentes . . . . .	188
C.2. Ejemplo de diagrama de actividades . . . . .	190
C.3. Ejemplo de diagrama de caso de uso . . . . .	191
C.4. Ejemplo de diagrama de secuencia . . . . .	193
D.1. Distribución de los pines del módulo WiFi . . . . .	196



# Índice de Tablas

3.1. Formación de la cadena de búsqueda para búsquedas automáticas.	29
3.2. Criterios de extracción de datos. . . . .	32
3.3. Número de estudios primarios obtenidos en el proceso de selección.	33
3.4. Evaluación de la calidad de los estudios primarios. . . . .	33
3.5. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ1. . . . .	34
3.6. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ3. . . . .	37
3.7. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ4. . . . .	39
3.8. Porcentajes individuales para el criterio de extracción de la sub-pregunta RQ5. . . . .	40
4.1. Descripción de atributos de la estructura de los perfiles de usuario propuesta. . . . .	61
5.1. Descripción de URLs definidas por el Middleware de Interacción.	92
5.2. Descripción de URLs definidas por el Controlador de ejecución.	93
6.1. Cuestionario para medir variables de percepción. . . . .	119
6.2. Preguntas abiertas del cuestionario. . . . .	120
6.3. Variables dependientes basadas en la percepción. . . . .	124
6.4. Variables dependientes basadas en el rendimiento. . . . .	124
6.5. Niveles de significancia [56]. . . . .	127
6.6. UC1 - Resultados del análisis de percepciones del usuario. . . .	128
6.7. UC1 - Estadística descriptiva para variables basadas en el Rendimiento del Usuario. . . . .	129
6.8. UC1 - Regresión Simple entre la Eficiencia Actual y la Facilidad de Uso Percibida. . . . .	129





6.9. UC1 - Regresión Simple entre la Efectividad Actual y la Utilidad Percibida. . . . .	130
6.10. UC1 - Regresión Simple entre la Facilidad de Uso Percibida y la Utilidad Percibida. . . . .	131
6.11. UC1 - Regresión Simple entre la Facilidad de Uso Percibida y la Intención de Uso. . . . .	131
6.12. UC1 - Regresión Simple entre la Utilidad Percibida y la Intención de Uso. . . . .	132
6.13. UC2 - Resultados del análisis de percepciones del usuario. . . .	133
6.14. UC2 - Estadística descriptiva para variables basadas en el Rendimiento del Usuario. . . . .	134
6.15. UC2 - Regresión Simple entre la Eficiencia Actual y la Facilidad de Uso Percibida. . . . .	135
6.16. UC2 - Regresión Simple entre la Efectividad Actual y la Utilidad Percibida. . . . .	135
6.17. UC2 - Regresión Simple entre la Facilidad de Uso Percibida y la Utilidad Percibida. . . . .	136
6.18. UC2 - Regresión Simple entre la Utilidad Percibida y la Intención de Uso. . . . .	137
6.19. UC2 - Regresión Simple entre la Facilidad de Uso Percibida y la Intención de Uso. . . . .	137
6.20. Tabla de resumen de estadísticos descriptivos. . . . .	138
6.21. Tabla de resumen de medias de la Facilidad de Uso Percibida (PEOU). . . . .	140
6.22. Tabla de resumen de medias de la Utilidad Percibida (PU). . .	141
6.23. Tabla resumen de medias de la Intención de Uso (ITU). . . .	141
6.24. Tabla resumen de aceptación de hipótesis de los cuasi-experimentos.	142
A.1. Estudios primarios seleccionados de IEEE Xplore . . . . .	166
A.2. Estudios primarios seleccionados de ACM Digital Library . . .	168
A.3. Estudios primarios seleccionados de Science Direct . . . . .	169
A.4. Estudios primarios seleccionados de Springer Link. . . . .	181
B.1. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ1 . . . . .	182
B.2. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ2 . . . . .	184
B.3. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ3 . . . . .	186
B.4. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ4 . . . . .	186



B.5. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ5 . . . . .	187
B.6. Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ6 . . . . .	187
C.1. Elementos utilizados en un diagrama de componente . . . . .	189
C.2. Elementos utilizados en un diagrama de actividades . . . . .	190
C.3. Elementos utilizados en un diagrama de caso de uso . . . . .	192
C.4. Elementos utilizados en un diagrama de secuencia . . . . .	194
E.1. Resultados efectividad de egresados/profesionales . . . . .	217
E.2. Resultados eficiencia de egresados/profesionales . . . . .	217
E.3. Resultados efectividad de estudiantes . . . . .	218
E.4. Resultados eficiencia de estudiantes . . . . .	219



### Cláusula de Propiedad Intelectual

---

Yo, Fernando Xavier Garnica Bautista, autor del trabajo de titulación "Creación de una arquitectura enfocada en la interacción hombre-computador para aplicaciones de internet de las cosas en ambientes de vida asistidos", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 31 de octubre del 2018

Una firma manuscrita en tinta azul sobre una línea horizontal.

Fernando Xavier Garnica Bautista

C.I: 0105890404



#### Cláusula de Propiedad Intelectual

---

Yo, Edwin Xavier Maita Tepán, autor del trabajo de titulación "Creación de una arquitectura enfocada en la interacción hombre-computador para aplicaciones de internet de las cosas en ambientes de vida asistidos", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 31 de octubre del 2018

Una firma manuscrita en tinta azul que parece decir "Edwin Maita Tepán".

Edwin Xavier Maita Tepán

C.I.: 0105661326



Cláusula de licencia y autorización para publicación en el Repositorio  
Institucional

---

Yo, Fernando Xavier Garnica Bautista, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Creación de una arquitectura enfocada en la interacción hombre-computador para aplicaciones de internet de las cosas en ambientes de vida asistidos", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 31 de octubre del 2018



Fernando Xavier Garnica Bautista

C.I: 0105890404



Cláusula de licencia y autorización para publicación en el Repositorio  
Institucional

---

Yo, Edwin Xavier Maita Tepán, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Creación de una arquitectura enfocada en la interacción hombre-computador para aplicaciones de internet de las cosas en ambientes de vida asistidos", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 31 de octubre del 2018

Edwin Xavier Maita Tepán

C.I.: 0105661326



# Dedicatoria

A mis padres, Carlos y Evita, por ser la motivación de mi vida, mis pilares para seguir adelante, por enseñarme valores que me han llevado a alcanzar esta gran meta.

A Enrique, Zoila, Gloria y Hugo, por el apoyo, por el cariño y por la unión que nos caracteriza con la cual, día a día, demostramos al mundo el verdadero significado de la palabra “hermanos”.

A Mónica, por el apoyo, comprensión, motivación y confianza que has sabido darme en cada momento durante esta bella etapa de mi vida.

Fernando Xavier Garnica Bautista



# Dedicatoria

Mis padres, Angel y Maria por ser el pilar fundamental, en toda mi educación, tanto académica, como de la vida, por su amor incondicional, trabajo y sacrificio en todos estos años para llegar a cumplir esta meta.

A mis hermanos, Geovanny, Jessica y Anthony que representan un ejemplo de perseverancia y superación, por su apoyo y consejos permitiendome aprender cada día de ellos.

Finalmente a mi compañero de tesis, Xavier por compartir experiencias, tiempo, trabajo y sobre todo por su amistad y confianza para lograr esta meta.

Edwin Xavier Maita Tepán





# Agradecimientos

Queremos expresar nuestro más profundo agradecimiento a la Ing. Priscila Cedillo Orellana, PhD. por sus enseñanzas enriquecedoras, por brindarnos su conocimiento y las directrices necesarias para realizar este trabajo, por su increíble amabilidad y su amistad.

A nuestros profesores, quienes con la enseñanza de sus valiosos conocimientos hicieron que podamos prepararnos de la mejor manera para enriquecer el contenido de este trabajo.

A nuestra familia, amigos, compañeros y a todas aquellas personas que nos brindaron de su tiempo, paciencia, dedicación y apoyo para que podamos alcanzar esta gran meta.

Fernando Xavier Garnica Bautista  
Edwin Xavier Maita Tepán



# Capítulo 1

## Introducción

Este capítulo presenta la introducción al presente trabajo de titulación y se distribuye de la siguiente manera: en la sección 1.1 se presenta la motivación y el contexto en el que se ha llevado a cabo el desarrollo de este trabajo de titulación; la sección 1.2 expone la problemática actual de las personas a las que va dirigida la propuesta desde el punto de vista humano y tecnológico; la sección 1.3 presenta las hipótesis y los objetivos de la investigación; la sección 1.4 muestra una breve aproximación al contenido de los capítulos presentados en este trabajo; finalmente, en la sección 1.5 se da a conocer la metodología de investigación adoptada para la elaboración de la propuesta.



## 1.1. Motivación y contexto

En la actualidad, existen varios sectores vulnerables, los mismos que se consideran prioritarios tanto dentro de las políticas de los gobiernos, como también dentro del área científica. La tecnología no aparece y evoluciona únicamente con el fin de sorprender, más bien, su enfoque debe estar centrado en brindar soluciones que faciliten la realización de actividades cotidianas, no sólo de los usuarios comunes, sino también de aquellas personas que no pueden interactuar con las diferentes aplicaciones tecnológicas de la misma manera en la que lo haría una persona que posee todas sus facultades, tanto físicas como mentales [72].

Entre los sectores vulnerables, se encuentran los adultos mayores y las personas con discapacidad [87] ya que, para estos grupos humanos, existen mayores limitaciones de acceso a recursos de subsistencia y se incrementan las necesidades de atención. Según el último informe mundial sobre discapacidad publicado por la Organización Mundial de la Salud (OMS), existen más de mil millones de personas en el mundo que poseen algún tipo de discapacidad, esta cantidad representa al 15 % de la población mundial [82]. Por su parte, el Consejo Nacional para la Igualdad de Discapacidades (CONADIS), establece que en Ecuador alrededor de 443.000 personas poseen algún tipo de discapacidad, ya sea física, intelectual, auditiva, visual, de lenguaje, psicológicas o psicosociales [22]. En la Figura 1-1 se puede apreciar un diagrama que indica el porcentaje de personas con discapacidad en el Ecuador según el tipo de discapacidad.

Con respecto a los adultos mayores, según los datos recogidos por las Naciones Unidas, en el año 2017 se registró la cantidad de 962 millones de personas mayores a 60 años, lo cual representa al 13 % de la población mundial [60]; se prevé que, en el 2050, esta cifra se duplicará [58]. Según el Instituto Nacional de Estadísticas y Censos (INEC), en Ecuador, el término “Adulto Mayor” se refiere a las personas de más de 65 años, mientras que el término “vejez” se ha dejado de usar por considerarse peyorativo y humillante [39]. En nuestro país existen 1.049.824 personas mayores de 65 años, esto representa al 6,5 % de la población total; además, se prevé que en el año 2020 será del 7,4 % [52].

Según el CONADIS, en Ecuador el 47 % de discapacitados posee una discapacidad física (Figura 1.1) [22] y el 6,5 % de personas son adultos mayores; ante esta perspectiva, surge la expresión “personas con movilidad reducida” la cual se refiere a aquellas personas con discapacidades físicas y adultos mayores que poseen muy poca o nula capacidad de llevar a cabo actividades en donde la fuerza física sea necesaria. A estas personas se les dificulta llevar a cabo ciertas actividades cotidianas, no poseen autonomía, confianza y movilidad volviéndose dependientes de sus familiares o personas a su cargo [71].

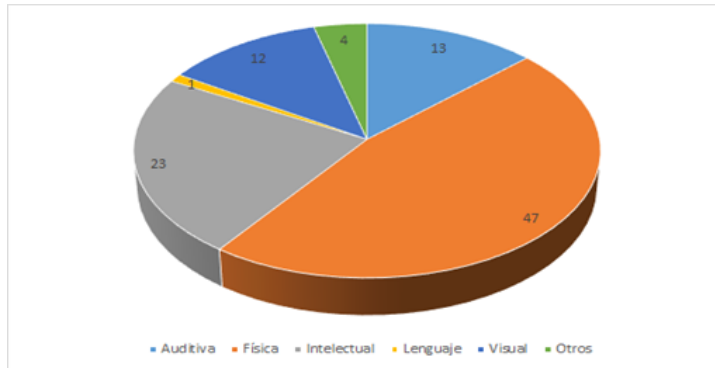


Figura 1.1: Personas con discapacidad en Ecuador según el tipo de discapacidad. Fuente: [22].

La tecnología puede contribuir enormemente a la consecución de los objetivos mencionados, es por ello que se han creado paradigmas como Vida Asistida por el Entorno (AAL, por sus siglas del inglés, *Ambient Assisted Living*), la cual se refiere a la aplicación de la inteligencia ambiental con la finalidad de promover y prolongar la vida independiente de los adultos mayores y de las personas con discapacidad, aplicando las Tecnologías de Información y Comunicaciones (TIC) de manera que éstas interactúen con las personas de forma natural y transparente en el lugar y contexto donde el usuario las necesite [31]. Como ejemplos de sistemas que forma parte de soluciones para AAL, se puede encontrar sistemas para seguimiento a distancia de signos vitales [59], sistemas para la detección de situaciones de desamparo en el hogar [15], entre otros.

Además, el paradigma de Internet de las Cosas (IoT, por sus siglas del inglés *Internet Of Things*), permite la interacción entre la persona y diferentes “cosas” de la vida cotidiana interconectados entre sí. Estas “cosas” (electrodomésticos, sensores etc.) se comunican entre ellas para crear servicios que faciliten tareas en el hogar, a partir de la detección de algún evento físico, sin la necesidad de la intervención humana directa. Algunas de las soluciones que involucran IoT y AAL son: dispositivos personales para la gestión de la terapia de diabetes [43], arquitectura para la movilidad y seguridad en ambientes médicos que brinda una conectividad del paciente con sensores y todo lo que le rodea [79], pastillero inteligente para la toma de medicamentos de adultos mayores [61], entre otras.



## 1.2. Planteamiento del problema

La necesidad de brindar a las personas una interacción más natural con aplicaciones y/o dispositivos usados en entornos de vida asistidos, es un aspecto que debe ser considerado de manera prioritaria durante la elicitación de requerimientos y que, durante la etapa de desarrollo, debe ser implementada minuciosamente, puesto que las interacciones representan el canal de comunicación directo que van a tener las personas con tales aplicaciones. En el contexto de AAL, la interacción se produce entre las soluciones IoT y los usuarios que pueden ser adultos mayores y/o personas con distintos tipos de discapacidades. Estos usuarios, por lo general, no poseen todas las facultades necesarias para interactuar con las aplicaciones de éstos entornos, debido a ciertas limitaciones propias de su situación. Tales inconvenientes provocan que cada uno tenga disponible únicamente ciertas formas de interacción: usando el movimiento de sus ojos, mediante el habla, a través de gestos o, en casos extremos, el pensamiento, es decir, los usuarios necesitan formas de interacción que se adapten a sus habilidades y los ayuden a superar las barreras físicas y cognitivas [2].

En la literatura se han encontrado diferentes propuestas de arquitecturas para aplicaciones IoT en entornos AAL [65][42][28][89]; sin embargo, cada una de ellas se enfoca en un sólo tipo de interacción para un grupo de usuarios muy específico, dejando de lado la posibilidad de que, bajo el mismo ambiente, puedan convivir varias personas con distintas necesidades de interacción. Por ejemplo, en [65] se presenta una arquitectura para un sistema que representa a un cuidador virtual que se despliega en hogares inteligentes y cuya finalidad es apoyar las actividades de la vida diaria de los adultos mayores a través de una interacción gestual. Por otro lado, en [42] se presenta una arquitectura para aplicaciones que ayudan a traducir las señales cerebrales a comandos que interactúan con el entorno mediante redes de actuación IoT, ejecutando las acciones pensadas por el usuario. Además, en [28] se da a conocer una arquitectura de un sistema domótico, éste sistema controla el ambiente usando comandos de voz. Finalmente, en [89] se presenta una arquitectura de un sistema de monitoreo de ambiente inteligente para recopilar datos de las condiciones físicas de los empleados y las condiciones del aire del entorno del trabajo en donde laboran, lo cual hace referencia a una interacción basada en sensores.

AAL puede solventar varios de los problemas presentados por personas con movilidad reducida, así como también IoT puede crear diversas soluciones que interactúen con los usuarios [73]; sin embargo, de lo que se ha indagado, hace falta una arquitectura específica, que permita hacer que estas áreas converjan de una manera integrada y centrada en conseguir una interacción natural con los usuarios que pertenecen a sectores vulnerables específicos: adultos mayores y personas con discapacidad.



## 1.3. Solución propuesta

La solución propuesta en este trabajo de titulación, busca el aporte de una arquitectura que sirva de soporte durante la construcción de soluciones que faciliten la interacción entre usuarios con necesidades especiales y diferentes tipos de dispositivos, sensores o actuadores conectados a Internet dentro de entornos de vida asistida, de tal manera que contribuyan a mejorar su calidad de vida.

## 1.4. Hipótesis y objetivos

En esta sección se presentan la hipótesis nula y alternativa; además, se da a conocer el objetivo general y los objetivos específicos.

### 1.4.1. Hipótesis

$H_0$ : La arquitectura de software propuesta, no brinda soporte en la creación de interacciones naturales de usuario en soluciones de Internet de las Cosas para ambientes de vida asistidos.

$H_1$ : La arquitectura de software propuesta, sí brinda soporte en la creación de interacciones naturales de usuario en soluciones de Internet de las Cosas para ambientes de vida asistidos.

### 1.4.2. Objetivos

En la siguiente sub-sección se establecen los objetivos tanto general como específicos del presente trabajo de titulación.

#### 1.4.2.1. Objetivo general

Diseñar una arquitectura para establecer diferentes tipos de interacciones hombre-computador (HCI) en ambientes de vida asistidos (AAL) que incluyen soluciones de Internet de las Cosas (IoT).

#### 1.4.2.2. Objetivos específicos

- Analizar el estado actual de la investigación sobre las técnicas y estrategias que se utilizan para el desarrollo e implementación de sistemas basados en AAL, HCI e IoT.
- Diseñar una arquitectura que integre los módulos y componentes necesarios para permitir la interconexión de tecnologías de forma adecuada en



un mismo sistema considerando el entorno o escenario de uso al que será destinada.

- Realizar la implementación de un prototipo de un sistema que haga uso de los componentes de la arquitectura de software propuesta.
- Evaluar la arquitectura propuesta, mediante un conjunto de escenarios, utilizando métodos de Ingeniería de Software, que permitan medir la inclusión de interacciones naturales en sistemas para ambientes de vida asistidos, por parte de personas afines al dominio de la Ingeniería de Sistemas.

## 1.5. Metodología de la investigación

Para la elaboración de la propuesta, se ha seguido una metodología estructurada conforme al modelo de transferencia tecnológica presentada en [32]. Este modelo de transferencia tecnológica contempla ocho actividades que permiten encontrar una solución realista mediante un proceso iterativo de validación empírica de un conjunto de soluciones candidatas. A continuación, se detallan cada una de las ocho actividades:

1. Análisis del problema: es necesario comprender el problema que da origen y propósito a la investigación, para ello se debe observar el dominio e identificar las necesidades de las partes interesadas.
2. Formulación del problema: al finalizar el análisis del problema, se lo debe formular de forma clara y precisa, incluyendo factores de contexto, los objetivos perseguidos por la investigación, planteamiento de preguntas de investigación y la respectiva justificación del estudio.
3. Revisión del estado del arte: mediante una revisión o mapeo sistemático de la literatura se esclarece el estado del arte, revisando soluciones existentes e identificando brechas que la investigación desea abordar.
4. Solución candidata: se propone una solución al problema establecido, por medio de un método definido.
5. Entrenamiento: se refiere a llevar a cabo una actividad de tipo incremental, mediante la cual busca proporcionar el conocimiento necesario a los profesionales del área para generar una vista general de la aplicación de la solución propuesta.
6. Validación inicial: la validación de la solución propuesta se la realiza en un entorno de laboratorio, para lo cual se utiliza un caso de estudio.

7. Validación realista: se aplica un entorno real industrial para llevar a cabo casos de estudio o experimentos controlados. Este paso se dejará como parte de trabajo futuro, puesto que requiere de ambientes reales y comités de ética.
8. Liberación de la solución: Se hace una valoración de los resultados obtenidos y se preparan las herramientas y material requerido para su despliegue y uso posterior. Al ser posterior a la validación realista, este paso se deja planteado como trabajo futuro.

En la Figura 1.2 se presentan cada una de las actividades propuestas por la metodología y su flujo iterativo. Cabe mencionar que en este trabajo de titulación se han cubierto las primeras seis actividades, proponiendo como trabajos futuros las actividades de validación realista y liberación de la solución.

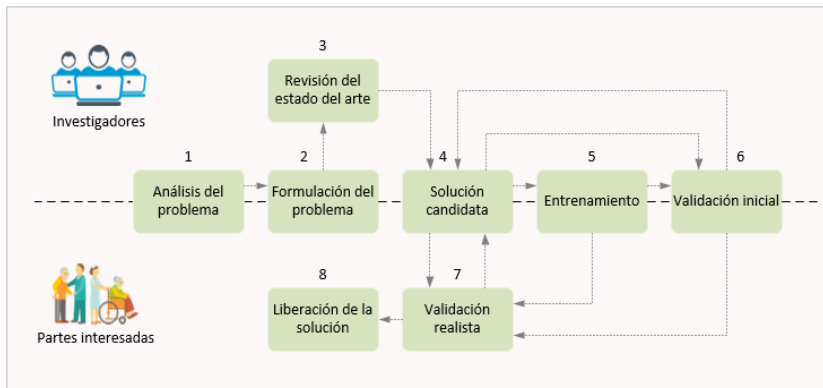


Figura 1.2: Metodología de la investigación. Fuente: Elaboración propia.

## 1.6. Estructura del trabajo

A continuación, se presenta la estructura que posee el presente trabajo de titulación dando una breve descripción de cada uno de los capítulos.

- Capítulo 1: Introducción.

Presenta la motivación, el contexto, la problemática actual, la solución propuesta, incluyendo la hipótesis, el objetivo general y los objetivos específicos de este trabajo de titulación. Este capítulo contrasta con las actividades número uno y dos de la metodología adoptada, esto debido a que expone de forma clara el análisis y la formulación del problema.





- **Capítulo 2: Marco tecnológico.**  
Presenta los conceptos correspondientes a los términos principales que son necesarios para la comprensión del presente trabajo de titulación, puesto que introducen ideas generales de las diferentes tecnologías que forman parte de la propuesta. La elaboración de este capítulo forma parte de la actividad número tres de la metodología adoptada ya que en esta se lleva a cabo la revisión del estado del arte.
- **Capítulo 3: Estado del arte.**  
El estado del arte se presenta mediante la realización de un mapeo sistemático de la literatura que tiene como objetivos la búsqueda, identificación y análisis de diversos estudios primarios publicados en revistas y conferencias digitales relacionados al dominio en el que se enfoca el presente proyecto de investigación. El desarrollo de este capítulo, al igual que el capítulo anterior, forma parte de la actividad número tres de la metodología adoptada.
- **Capítulo 4: Arquitectura de software propuesta.**  
Presenta la arquitectura de software propuesta dando a conocer la descripción y el funcionamiento de cada uno de sus componentes desde un punto de vista estático, funcional y dinámico. Este capítulo contrasta con la actividad número cuatro de la metodología adoptada ya que es en ésta en donde se propone la solución candidata.
- **Capítulo 5: Implementación.**  
Aquí se lleva a cabo la implementación de un sistema basado en la arquitectura de software propuesta, dando a conocer los diferentes módulos que lo componen y las diferentes entradas y salidas de cada proceso. Este capítulo forma parte de la actividad número cinco de la metodología propuesta.
- **Capítulo 6: Validación de la propuesta.**  
Presenta la validación de la arquitectura de software propuesta en el capítulo 4 sobre la implementación presentada en el capítulo 5. Este capítulo forma parte de la actividad número 6 de la metodología adoptada.
- **Capítulo 7: Conclusiones y trabajos futuros.**  
Se presenta la información más relevante obtenida con el desarrollo de este trabajo de titulación. Además, se proponen trabajos futuros relacionados al presente.
- **Apéndices.**  
Presenta la documentación adjunta y generada durante el desarrollo del

presente trabajo de titulación, por ejemplo: información de los artículos de investigación aceptados en diferentes conferencias, resultados completos del mapeo sistemático, diagramas, entre otros.

La Figura 1.3 muestra un esquema de la estructura del trabajo de titulación, y relaciona las actividades de la metodología adoptada con cada capítulo.

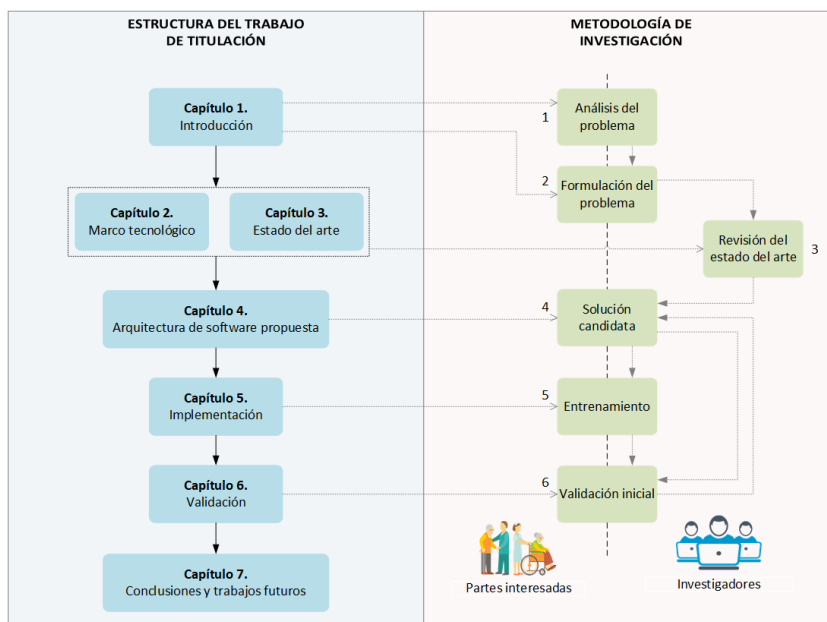


Figura 1.3: Estructura del trabajo de titulación. Fuente: Elaboración propia.

## 1.7. Aporte científico

Este trabajo de titulación se ha realizado en el contexto del proyecto de investigación: "Fog Computing aplicado a monitoreo de dispositivos usados en ambientes de vida asistidos (*Ambient Assisted Living*); caso de estudio: plataforma para el adulto mayor", aportando como un estudio preliminar del mismo. Además, muestra su aporte científico mediante un artículo de difusión científica, elaborado y enviado a la conferencia INCISCOS 2018, en espera de respuesta por parte de pares académicos:

- "Towards an Architecture for Human Computer Interactions in the Internet of Things. Ambient Assisted Living Domain" (enviado).





## Capítulo 2

# Marco tecnológico

En este capítulo se presentan los conceptos correspondientes a los términos principales abordados en el presente trabajo de titulación. Dichos conceptos facilitan el entendimiento de los temas abordados a lo largo de los diferentes capítulos e introducen ideas generales de las diferentes tecnologías que forman parte de la propuesta.

El capítulo se distribuye de la siguiente manera: en la sección 2.1 se define la Vida Asistida por el Entorno con sus objetivos y de la inteligencia ambiental; la sección 2.2 expone la definición de Internet de las Cosas, sus características, descripción del modelo de referencia y los dispositivos; la sección 2.3 presenta la definición de Interacción Humano-Computadora, sus fundamentos, paradigmas y estilos de interacción; la sección 2.4 presenta el significado de e-Health; finalmente, en la sección 2.5 se describe el significado de arquitectura de software y las diferentes vistas que posee.



## 2.1. Vida asistida por el entorno

Tomando en cuenta el contexto social, los adultos mayores procuran llevar a cabo sus actividades cotidianas de forma independiente el mayor tiempo posible tratando de valerse por sí mismos y no depender de familiares o de personas a su cargo. Además, las personas con diversidad funcional deben gozar también de cierta libertad para llevar a cabo algunas de sus actividades cotidianas, consiguiendo que logren cierta independencia y que su autoestima vaya en aumento [26]. En este contexto, es necesario también hacer énfasis en aquellas personas que padecen enfermedades crónicas, o bien, están en proceso de recuperación, las cuales necesitan de un constante monitoreo y cuidado tanto para prolongar su tiempo de vida como para servir de apoyo en el proceso de recuperación [21]. Por otro lado, en el ámbito tecnológico, se ha notado un avance gigantesco en Tecnologías de la Información, Comunicación y Control (TICC) las cuales han permitido el fácil acceso a internet, telefonía móvil, domótica, nuevos sistemas de interacción humano computadora, entre otros [16]. En la actualidad todas estas tecnologías pueden ser empleadas en conjunto para promover una vida y envejecimiento independientes el mayor tiempo posible.

Considerando las necesidades de los adultos mayores y personas con diversidad funcional en conjunto con los avances en las TICC surge el concepto de Vida Asistida por el Entorno (AAL, del inglés Ambient Assisted Living) cuyo enfoque se refiere a la forma de garantizar mejores condiciones de vida tanto para adultos mayores, personas con diversidad funcional, personas con enfermedades crónicas y en estado de recuperación de dichas enfermedades mediante el desarrollo de tecnologías y servicios innovadores [49]. La tecnología usada en AAL ha sido diseñada para que las personas mayores y con diversidad funcional, sean capaces de vivir de forma independiente, ayudándoles a realizar tareas cotidianas que, por lo general, no pueden realizarlas adecuadamente por sí mismas [47]. De esta forma, se crea un entorno en el que las personas puedan continuar siendo ciudadanos autosuficientes.

### 2.1.1. Inteligencia ambiental

La Inteligencia Ambiental es un paradigma en las tecnologías de la información en los sitios en donde se encuentran los usuarios, para que, interactúen con ellos de forma natural, transparente, sensible y adaptable a sus necesidades. Para conseguirlo, todo el entorno que los rodea, hogares y oficinas, automóviles y ciudades, desarrollará colectivamente una red generalizada de dispositivos inteligentes que se agrupan, procesan y transportan la información, adaptándose a las necesidades y el comportamiento del usuario para mejorar su calidad de vida [70].



La Inteligencia Ambiental cuenta con tres tecnologías fundamentales las cuales son: computación ubicua, comunicación ubicua e interfaces inteligentes [66]. A continuación, se da una breve descripción de cada uno de ellos.

- Computación ubicua: mediante la dotación de procesamiento y comunicación a los objetos cotidianos y la interconexión de sistemas embebidos camuflados en el entorno.
- Comunicación ubicua: todos los dispositivos que forman parte de un entorno inteligente propio de la computación ubicua deberán interactuar entre sí y con el usuario. Para ello es necesario el desarrollo de una comunicación ubicua que permita al usuario acceder a los servicios computacionales ofrecidos desde cualquier lugar
- Interfaces inteligentes: Las interfaces inteligentes permiten al usuario comunicarse con los dispositivos del entorno inteligente de una manera sencilla y natural. El objetivo principal de estas interfaces es ocultar al usuario la complejidad del sistema y sólo mostrarle sus funcionalidades

### 2.1.2. Objetivos de Vida Asistida por el Entorno

Se pueden listar una gran cantidad de objetivos que la tecnología usada en AAL debe cumplir, aquí se presentan los más importantes [1]:

- Incrementar el tiempo de vida de las personas en el entorno deseado mejorando su autonomía, confianza y movilidad.
- Apoyar y promover el mantenimiento de la salud con un mejor estilo de vida para personas en situación de riesgo.
- Mejorar la seguridad y prevenir el aislamiento social de las personas en situación de riesgo.
- Brindar a los cuidadores y familiares apoyo en el cuidado y monitoreo de personas en situación de riesgo.
- Mejorar la eficiencia y productividad de los recursos tecnológicos empleados en los diferentes entornos.

## 2.2. Internet de las cosas

La Unión Internacional de Telecomunicaciones (ITU) define al Internet de las cosas (IoT, por sus siglas del inglés Internet of things) como una infraestructura mundial para la sociedad de la información, que propicia la prestación de



servicios avanzados mediante la interconexión de objetos (físicos y virtuales), gracias a la interoperabilidad de tecnologías de la información (TIC) y la comunicación presentes y futuras. Aprovechando las capacidades de identificación, adquisición de datos, procesamiento y comunicación, IoT utiliza plenamente los “objetos” para ofrecer servicios a todos los tipos de aplicaciones, garantizando a su vez el cumplimiento de los requisitos de seguridad y privacidad [40].

### 2.2.1. Características

- Interconectividad. En el contexto de IoT, todo puede estar interconectado con la infraestructura mundial de la información y la comunicación [40].
- Servicios relacionados con objetos. IoT es capaz de suministrar servicios relacionados con los objetos dentro de las restricciones de objetos, como protección de la privacidad y coherencia semántica entre los objetos físicos y sus correspondientes objetos virtuales. Para ofrecer servicios relacionados con objetos dentro de las restricciones de objetos, las tecnologías en el mundo físico y en el de la información evolucionarán [40].
- Heterogeneidad. Los dispositivos en IoT son heterogéneos dado que se basan en diferentes plataformas hardware y redes. Pueden interactuar con otros dispositivos o plataformas de servicios a través de redes diferentes [40].
- Cambios dinámicos. El estado de los dispositivos varía dinámicamente, por ejemplo, del modo reposo al activo, conectado y/o desconectado, así como el contexto del dispositivo, como la ubicación y velocidad. Además, el número de dispositivos también puede cambiar dinámicamente [40].
- Escala enorme. El número de dispositivos que se gestionan y que se comunican entre sí puede ser incluso un orden de magnitud mayor que el número de dispositivos conectados actualmente a Internet. El porcentaje de comunicación que requerirán estos dispositivos será muchísimo mayor que el de la comunicación entre humanos. Será incluso más esencial la gestión de los datos generados y su interpretación para fines de aplicación, aspectos éstos que guardan relación con la semántica de datos y la manipulación eficiente de datos [40].

### 2.2.2. Modelo de referencia

Según [40], el modelo consta de cuatro capas y de capacidades de gestión, y de seguridad relacionadas a las capas como se muestra en la Figura 2.1.

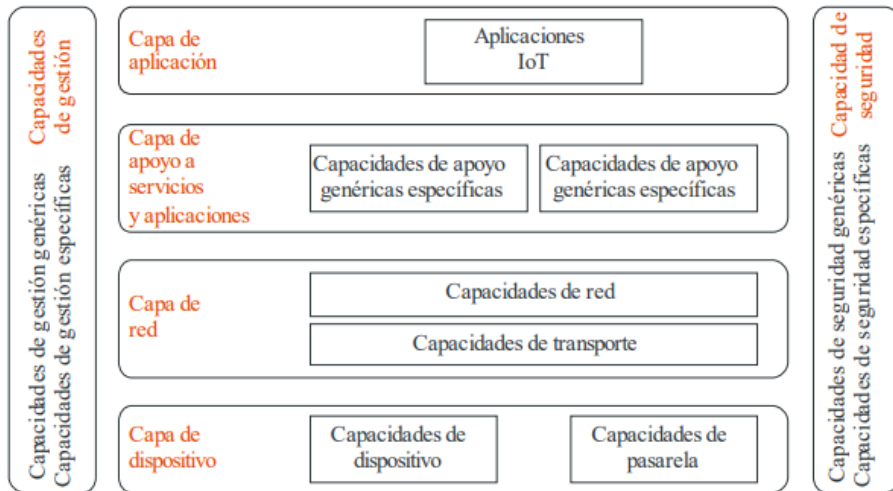


Figura 2.1: Modelo de referencia. Fuente: [40].

A continuación se describe cada capa del modelo de referencia:

- **Capa de aplicación.** Esta capa contiene las aplicaciones IoT.
- **Capa de soporte de servicios y aplicaciones.** Se describe en términos de las siguientes capas:
  - Genérica: Son capas comunes que puede utilizarlas diferentes aplicaciones IoT, tales como procesamiento o almacenamiento de datos.
  - Específica: Son capas que sirven para atender las necesidades particulares de diversas aplicaciones.
- **Capa de red.** Esta capa se subdivide en:
  - Capa de red: Funciones de control de la conectividad en red. Funciones de control de acceso y de recursos de transporte.
  - Capa de transporte: enfocadas en suministrar conectividad para la transmisión de datos e información específica para servicios y aplicaciones.
- **Capa de dispositivo.** Está subdividida en las siguientes capas:
  - Capa de dispositivo: interacción directa con la red (sin pasarela), interacción indirecta (con pasarela) con la red de comunicación, redes ad-hoc, modo reposo y activo (ahorro de energía).





- Capa de pasarela: soporte de interfaces múltiples, diferentes dispositivos conectados mediante diferentes tipos de tecnologías alámbricas e inalámbricas. La conversión de protocolo se utiliza cuando las comunicaciones en la capa de dispositivo utilizan protocolos diferentes y la otra es cuando en la comunicación intervienen la capa de dispositivo y la de red y se utilizan protocolos.
- **Capa de gestión.** Se encarga de la gestión de fallos, configuración, contabilidad, rendimiento y seguridad. Se clasifican en:
  - Genéricas: Gestión de dispositivos como activación y desactivación de dispositivos remotos, diagnósticos, actualizaciones, estado de dispositivos. Gestión del tráfico y congestión como a detección de las condiciones de saturación de red y la gestión de la topología de red.
  - Específicas: dependerá de los requisitos particulares de cada aplicación.
- **Capa de seguridad.** Se divide en dos tipos:
  - Genéricas: Son independientes de la aplicación. Para la capa de aplicación confidencialidad de los datos y protección de la integridad entre otros. En la capa de red, autorización, autenticación, confidencialidad de datos de señalización. Y en capa de dispositivo, autenticación, autorización, validación de la integridad del dispositivo, control de acceso.
  - Específicas: Están relacionadas con los requisitos específicos de la aplicación.

### 2.2.3. Dispositivos

IoT consiste en conectar diferentes dispositivos al Internet para la transmisión y captura de información. Entre los dispositivos más usados están los sensores y los actuadores; a continuación, se brinda una descripción de cada uno.

#### 2.2.3.1. Sensores

Según [67]: un sensor es un dispositivo (típicamente electrónico) que detecta eventos o cambios en su entorno físico (por ejemplo, temperatura, sonido, calor, presión, flujo, magnetismo, movimiento y parámetros químicos y bioquímicos) y proporciona una salida correspondiente. El objetivo principal de un sensor es recopilar datos de su entorno y proporcionar salida a sus dispositivos contiguos



(por ejemplo, puertas de enlace y actuadores) o aplicaciones. Los sensores generalmente recolectan datos usando interfaces físicas (entradas) que detectan el entorno y luego convierten señales de entrada en señales eléctricas (salidas) que son entendidas por los dispositivos de comunicación e informática.

### 2.2.3.2. Actuadores

Según [67]: un actuador es un tipo de motor que se encarga de controlar o tomar medidas en un sistema. Toma una fuente de datos o energía (por ejemplo, presión de fluido hidráulico, otras fuentes de energía) y convierte los datos / energía en movimiento para controlar un sistema. Como se definió anteriormente, los sensores son responsables de detectar los cambios en su entorno, recopilar datos relevantes y poner dichos datos a disposición de los sistemas de monitoreo. Recopilar y mostrar datos mediante un sistema de monitoreo es inútil a menos que dichos datos se traduzcan en inteligencia que pueda usarse para controlar o gobernar un entorno antes de que se afecte un servicio. Los actuadores usan datos analizados y recolectados por el sensor, así como otros tipos de inteligencia de datos para controlar los sistemas de IoT. Por ejemplo, cerrar el flujo de gas cuando la presión medida está por debajo de un cierto umbral.

## 2.3. Interacción humano-computadora

La Interacción humano-computadora (HCI, por sus siglas en inglés Human-Computer Interaction) es un campo de estudio multidisciplinario que se centra en el diseño de la tecnología informática y, en particular, la interacción entre los seres humanos y las computadoras [76]. HCI hizo su aparición en la década de 1980 enfocándose únicamente en la interacción con las computadoras; actualmente, HCI se ha expandido para abarcar casi todas las formas de diseño de tecnología de la información.

HCI lleva a cabo procesos de diseño, evaluación e implementación de dispositivos tecnológicos interactivos, con el fin de estudiar todas las posibles interacciones que se puedan llegar a tener con tales dispositivos. El enfoque técnico original de HCI ha venido siendo la usabilidad, la cual, inicialmente, fue articulada con el lema “fácil de aprender, fácil de usar” [18]. Actualmente, la usabilidad de un sistema o dispositivo tecnológico es un punto clave para la consecución del objetivo principal de la interacción el cual se refiere a: conseguir un intercambio de información de forma eficiente, minimizando errores, incrementando la satisfacción, disminuyendo la frustración y haciendo más productivas las tareas que rodean a las personas y los computadores.



### 2.3.1. Fundamentos

Existen 3 aspectos o actores fundamentales que deben ser considerados en HCI, estos son: el humano o usuario, la computadora o sistema tecnológico y la interacción [25]:

- El humano: puede ser un usuario individual o un grupo de usuarios tratando de realizar un trabajo utilizando la tecnología.
- La computadora: cualquier tecnología, puede ser el ordenador general de sobremesa, un sistema informático a gran escala, o un sistema embebido.
- La interacción: cualquier comunicación entre un usuario y una computadora sea directa o indirecta.
  - Interacción directa: implica un diálogo con retroalimentación y control a lo largo de ejecución de la tarea.
  - Interacción indirecta: implica, por ejemplo, la implementación de sensores inteligentes que controlan el medio ambiente.

Existen dos cualidades interrelacionadas que un sistema tecnológico debe poseer [36]: útil y usable; sin embargo, [25] menciona una tercera: que el sistema sea usado. Cada una posee un fin:

- Útil: conseguir realizar la tarea propuesta.
- Usable: hacerlo con facilidad y de forma natural.
- Usado: hacer que la gente quiera usarlo.

Con la consecución de estas tres cualidades se asegura que el sistema tecnológico sea fácil de aprender, fácil de manejar y que presente un propósito reconocible.

### 2.3.2. Paradigmas y estilos de interacción

Los paradigmas de interacción hacen referencia a los modelos tecnológicos de los cuales se derivan los sistemas de interacción, entre los paradigmas más extendidos en la actualidad se pueden encontrar: Sistemas de ventanas e interfaces WIMP (Windows, Icons, Menus, Pointers).

- Computación ubicua.
- Realidad virtual.
- Realidad aumentada.



- Sensores y sensibilidad al contexto.

Existen varias formas de interactuar con un sistema tecnológico, estas formas de interacción vienen dadas por los estilos de interacción, los más frecuentes son:

- Línea de comandos.
- Interacción basada en interfaces WIMP y touch.
- Interacción asistida.
- Interacción basada en el movimiento.
- Interacción basada en audio.
- Interacción basada en la voz.
- Interacción basada en la visión.
- Interacción basada en sensores.

Tanto los paradigmas de interacción como los estilos de interacción en su mayor parte dependen de los avances tecnológicos y su aplicación creativa para mejorar la interacción. Por otro lado, cada estilo de interacción tiene como finalidad ayudar a un tipo de usuario específico a completar una actividad de la manera más natural posible [44], por ejemplo, aquellas personas con movilidad reducida pueden interactuar con un sistema mediante comandos de voz, comandos visuales, entre otros.

### 2.3.3. Interacción natural

Con la aparición de los sistemas de inteligencia ambiental, se está dejando de lado la interacción tradicional Persona-Computadora a través de interfaces gráficas, evitando que el usuario tenga que intervenir constantemente indicando al sistema cada una de las acciones que debe realizar [80] y se está empezando a adoptar interacciones semejantes a las interacciones Persona-Persona, la cual representa la manera más natural de interactuar y que, en poco tiempo, será vista como una interacción Persona-Ambiente [68].

La interacción natural de las personas con el ambiente se refiere a la ejecución de órdenes y comandos por parte del sistema inmerso en el contexto tomando como entradas las intenciones del usuario. El usuario puede solicitar la ejecución de dichas acciones mediante órdenes naturales, no necesariamente ligadas a la interacción con una pantalla mediante un puntero o señales táctiles, siendo capaz de usar gestos, la voz, la visión, pensamientos o el monitoreo de

sus signos vitales, actividades y del entorno a través de sensores [17]. Este tipo de interacción hace referencia a la interacción Persona-Computador implícita (iHCI) misma que en [74] se define como *“la interacción entre un humano con el entorno y con los artefactos ideados para lograr una meta, a través de un proceso de adquisición de entradas implícitas y obtención de salidas implícitas”*.

Considerando los sistemas AAL en los cuales intervienen personas que, en su mayoría, poseen diversidad funcional o son adultos mayores, las interacciones tradicionales, a través de interfaces gráficas de usuarios, tienden a ser tediosas debido a que no todos poseen las facultades necesarias para visualizar una pantalla, mover un puntero, o usar un dispositivo táctil [64]. Es por ello que las interacciones naturales adoptan un papel importante en entornos AAL, haciendo posible que cada usuario pueda usar la interacción que más se apegue a las capacidades que éste posee. A continuación, se describen algunas de las interacciones naturales más conocidas:

#### 2.3.3.1. Interacción basada en la voz

Permite la interacción entre una persona y un ordenador, dispositivo o sistema mediante la voz. Consiste en la ejecución de procesos o servicios en base a comandos vocales de entrada enviados por el usuario. Es muy importante que el sistema con el que se interactúa sea capaz de reconocer las órdenes recibidas y, en algunos casos, a la persona que lo hace, esto representa una tarea difícil pero que puede ser solventada con algoritmos de inteligencia artificial.



Figura 2.2: Interacción basada en la voz. Fuente: [5].

La mayoría de usuarios pueden acceder a este tipo de interacción, sin embargo, resulta ser una interacción complicada para las personas que no poseen la facultad de hablar o comunicarse a través de la voz debido a la dificultad de generar comandos usando palabras o sonidos.

### 2.3.3.2. Interacción basada en la visión

Mediante este tipo de interacción, los usuarios son capaces de controlar sistemas o dispositivos usando la visión. Esta interacción está basada en el seguimiento de la mirada del usuario (eyetracking); para conseguirlo, se puede evaluar el punto donde se fija la mirada midiendo los ángulos de la visión en relación a lo que se observa, o captando el movimiento y rotación del ojo en relación a la cabeza del usuario. Un eyetracker es un dispositivo que lleva a cabo el proceso de seguimiento de la mirada o la rotación de los ojos, el dispositivo puede ser un software que usa una videocámara que monitorea la mirada o gafas que miden el movimiento y rotación de los ojos. Dicho monitoreo puede ser realizado mediante luz infrarroja dirigida hacia la pupila, medición de reflexiones en la córnea usando una cámara, o detectando el objeto en el que una persona fija su mirada [63].

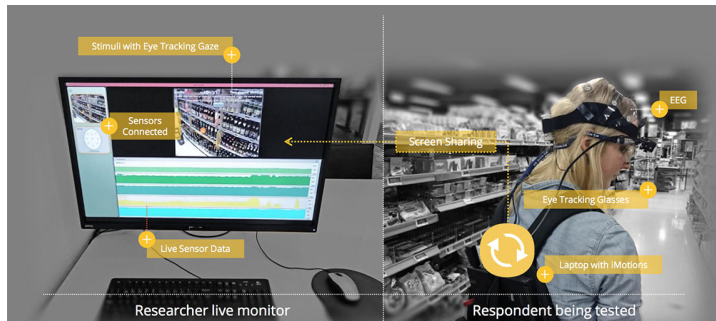


Figura 2.3: Monitoreo eyetracking. Fuente: [38].

Resulta difícil interactuar usando seguimiento visual para las personas que poseen problemas oculares, más específicamente, las personas que padecen ceguera. Sin embargo, es de gran utilidad para aquellas personas con movilidad reducida o aquellos que sufren de trastornos cognitivos, en general, la mayoría de personas pueden usar este tipo de interacción.

### 2.3.3.3. Interacción basada en comandos mentales

Permite interactuar con el ambiente usando el pensamiento humano. Este tipo de interacción se fundamenta con las Interfaces Cerebro-Computador (BCI, siglas del inglés Brain-Computer Interface) que se basa en la lectura y extracción de ondas cerebrales que posteriormente son procesadas e interpretadas por un computador. Según [85], BCI se refiere a la interacción con

dispositivos, donde las intenciones del usuario, representadas como varios estados del cerebro, se descifran y se traducen en acciones sin requerir ningún movimiento físico.



Figura 2.4: Control de un sistema robótico usando BCI. Fuente: [41].

Este tipo de interacción es ideal para aquellas personas que poseen enfermedades degenerativas que imposibilitan a una persona realizar movimientos musculares, tal es el caso de personas con Esclerosis Lateral Amiotrófica (ELA), o personas que han perdido sus extremidades, de manera que el único canal de comunicación posible con el ambiente es el cerebro.

#### 2.3.3.4. Interacción basada en gestos

Este tipo de interacción permite comunicarse con los dispositivos, sistemas y ordenadores empleando gestos o, en general, realizando movimientos corporales. Tales gestos suelen ser realizados mediante las manos o el rostro. El reconocimiento de los gestos es una tarea compleja que involucra aspectos como el modelado de movimiento, el análisis de movimiento, el reconocimiento de patrones y el aprendizaje automático [88].



Figura 2.5: Sistema controlado por gestos que se ejecuta en un televisor 3D. Fuente: [27].

La mayoría de personas pueden hacer uso de esta interacción, sin embargo, en entornos AAL, los mayores beneficiados son tanto los adultos mayores como las personas discapacitadas siempre que estas posean las facultades necesarias para realizar cualquier tipo de movimiento que sea reconocido por un dispositivo de reconocimiento gestual.

#### **2.3.3.5. Interacción basada en sensores**

Este tipo de interacción hace referencia al uso de diferentes sensores para monitorear, captar o medir valores ya sea del medio ambiente, del cuerpo humano, industriales, etc., con la finalidad de que un dispositivo, ordenador o el mismo sensor procese e interprete los valores recogidos y ejecute ciertas acciones en base a los resultados obtenidos.





Figura 2.6: Sistema basado en sensores para monitoreo cardíaco. Fuente: [81].

La interacción basada en sensores es accesible para cualquier tipo de usuario. Haciendo énfasis en el contexto AAL, son usados, típicamente, para monitorear los signos vitales de adultos mayores, enfermos crónicos, personas con discapacidad, niños, etc. [46]; además, se usan también para monitorear el medio ambiente [20], para sistemas de control de caídas [78], entre otros. Todos los valores captados son procesados e interpretados con la finalidad de generar alarmas de salud y enviar notificaciones a doctores, cuidadores o familiares.

## 2.4. e-Health

Según [55], e-health se refiere a un término necesario para describir el uso combinado de la comunicación electrónica y la tecnología de la información en el sector de la salud. Así como el uso de datos digitales transmitidos, almacenados y recuperados electrónicamente con fines clínicos, educativos y administrativos, tanto en el sitio local como a distancia.



## 2.5. Arquitectura de software

Según [37], *“la arquitectura de software es la suma de los módulos, procesos y datos no triviales del sistema, su estructura y relaciones exactas entre sí, cómo pueden y deben extenderse y modificarse, y de qué tecnologías dependen, desde cuál puede deducir las capacidades exactas y las flexibilidades del sistema, y desde cuál se puede formar un plan para la implementación o modificación del sistema”*. Las arquitecturas de software describen diversos aspectos del software, esta descripción es más comprensible cuando se realiza mediante modelos o vistas las cuales deben ser coherentes entre sí puesto que describen la misma cosa. Existen tres vistas fundamentales en cualquier arquitectura:

- **Vista estática:** describe los componentes que tiene la arquitectura.
- **Vista funcional:** describe qué hace cada componente.
- **Vista dinámica:** describe cómo se comportan los componentes a lo largo del tiempo y cómo interactúan entre sí.

Las diferentes vistas o modelos pueden ser representados mediante diferentes lenguajes. Existe un consenso en adoptar UML (por sus siglas en inglés, Unified Modeling Language) como lenguaje para describir todas las vistas o modelos mediante diagramas de actividades, diagramas de componentes, diagramas de flujo, entre otros.

### 2.5.1. Componente

Un componente es un bloque de construcción de software de cómputo. Según [35] define un componente como “una parte modular, desplegable y sustituible de un sistema, que incluye la implantación y expone un conjunto de interfaces”. Los componentes forman la arquitectura del software y, en consecuencia, juegan un papel en el logro de los objetivos y de los requerimientos del sistema que se va a construir. Un componente de software debe poseer las siguientes características:

- Ser reutilizable.
- Ser intercambiable.
- Poseer interfaces definidas.
- Ser cohesivos.



### 2.5.2. Módulo

Un módulo se define como una entidad de código fuente que encapsula datos y funciones que operan en datos (por ejemplo, clases, archivos de código fuente)[53]. Las interdependencias entre módulos son relaciones binarias que son compatibles con el lenguaje de programación utilizado para implementar el sistema (como llamadas de función/procedimiento, acceso a variables, herencia). Los programadores definen los límites del módulo agrupando grupos de funciones/métodos relacionados, macros y propiedades/variables en archivos o clases de códigos fuente. Las relaciones entre los módulos pueden obtenerse fácilmente extrayendo las relaciones a nivel de módulo del código fuente. A medida que los sistemas crecen, los ingenieros a menudo integran estos módulos en entidades de nivel superior llamadas subsistemas [54].





## Capítulo 3

# Estado del arte

En este capítulo, se presenta un mapeo sistemático de la literatura que tiene como objetivos la búsqueda, identificación y análisis de diversos estudios primarios publicados en revistas y conferencias digitales relacionados al dominio en el que se enfoca el presente trabajo de titulación. Los estudios de mapeo sistemático proporcionan una visión general de un área de investigación mediante contribuciones publicadas y categorías de clasificación definidas; además, en los mapeos sistemáticos, el análisis de evidencia es más amplio y las preguntas de investigación son más generales; finalmente, posibilitan la identificación de grupos o de vacíos de evidencia para detectar áreas donde deben realizarse más estudios primarios [62].

El mapeo sistemático propuesto en este capítulo, busca explorar el estado actual de las investigaciones en el contexto de las soluciones (sistemas, aplicaciones, implementaciones) basadas en IoT para el cuidado de la salud y cómo se lleva a cabo la interacción entre las personas y dichas soluciones. Se ha realizado una revisión de la literatura en búsqueda de estudios secundarios que aborden la temática expuesta, sin embargo, no se ha encontrado uno que responda las inquietudes planteadas en este trabajo de titulación.

El capítulo se distribuye de la siguiente manera: la sección 3.1 presenta la metodología adoptada para la realización del mapeo sistemático; la sección 3.2 presenta la etapa de planificación en donde se establecen las actividades para desarrollar el mapeo sistemático; la sección 3.3 muestra la etapa de ejecución de las actividades planteadas; la sección 3.4 expone la etapa de reporte que incluye los resultados obtenidos del mapeo sistemático y la sección 3.5 presenta las conclusiones del capítulo.



## 3.1. Metodología

Para llevar a cabo el mapeo sistemático se han seguido las recomendaciones de la metodología presentada por Kitchenham y Charters [45]. La metodología propone la realización de un conjunto de actividades específicas, tales actividades se dividen en tres etapas: i) Planificación, ii) Ejecución y iii) Reporte. Al completar las etapas, se obtendrá un resumen de la investigación de estudios primarios, basado en evidencia confiable, que permitirá conocer los beneficios y desventajas de la temática bajo estudio.

## 3.2. Etapa de planificación

La etapa de planificación establece cada una de las actividades que deben ser realizadas para el mapeo sistemático. Aquí se definen las preguntas de investigación, estrategias de búsqueda, amenazas de validez, período y cadena de búsqueda, criterios de extracción de datos, entre otros.

### 3.2.1. Preguntas de investigación

La pregunta de investigación que cubre el enfoque planteado es la siguiente: *"¿Cuáles son y bajo qué entornos funcionan las soluciones basadas en IoT dentro del dominio de la salud, a quiénes están dirigidas, cómo se recolecta la información para los diferentes tratamientos que cubren y que tipos de interacción ofrecen?"*.

Con la finalidad de dar respuesta a la pregunta de investigación general, se proponen las siguientes sub-preguntas que brindarán respuestas a aspectos más específicos:

- RQ1: ¿En qué entornos se aplican las soluciones basadas en IoT dentro del área de la salud y a quiénes están dirigidas?
- RQ2: ¿Cómo se lleva a cabo la recolección de datos fisiológicos y del entorno de una persona para el uso dentro del área de la salud?
- RQ3: ¿Qué aplicaciones, implementaciones, soluciones o sistemas propuestos basados en IoT existen para el área de la salud?
- RQ4: ¿Cuáles son las enfermedades, tratamientos y tipos de asistencia más relevantes que pueden ser cubiertos usando soluciones IoT en el área de la salud?
- RQ5: ¿Cuáles son las formas de interacción disponibles para el control de soluciones propuestas basadas en IoT en el área de la salud?



- RQ6: ¿De qué manera se ha llevado hasta ahora la investigación en estos temas?

### 3.2.2. Estrategia de búsqueda

Se han escogido cuatro librerías digitales para realizar la búsqueda automática, debido a que son librerías en donde se publican la mayor cantidad de proyectos relacionados a tecnología, al respecto, se hizo una búsqueda preliminar para establecer las librerías digitales que mayor cantidad de estudios primarios apegados a este tema incluyen y los que tuvieron mejores resultados en cuanto a impacto científico, estas son:

- *IEEE Xplore*
- *ACM Digital Library*
- *SpringerLink*
- *Science Direct*

Para las búsquedas manuales, se han extraído los estudios primarios más significativos de conferencias de renombre en donde las temáticas están bastante ligadas a la pregunta de investigación. Las conferencias elegidas son las siguientes:

- *International Conference on e-Health Networking, Applications and Services* (HealthCom)
- *International Conference on Smart Homes and Health Telematics* (ICOST)
- *International Work-Conference on Artificial Neural Networks* (IWANN)
- *European Conference on Ambient Intelligence* (AmI)
- *International Conference on Ubiquitous Computing and Ambient Intelligence* (UCAmI)

### 3.2.3. Cadena de búsqueda

Para apoyar la búsqueda automática, se ha definido un conjunto de palabras clave, las cuales se han relacionado mediante conectores lógicos, formando así una cadena de búsqueda que permitió obtener resultados muy apegados a nuestra temática. La Tabla 3.1 muestra la estructuración de la cadena de búsqueda mediante los conectores que unen los diferentes términos clave.



CADENA	SUBCADENA	CONECTOR	TÉRMINOS ALTERNATIVOS
Internet de las cosas	"IoT"	OR	"Internet of Things", "internet-of-things"
Internet de las cosas	"Internet Of Everything"	OR	
Internet de las cosas	"IoMT"	AND	"Internet Of Medical Things"
Salud	"e-health"	OR	"electronic health"
Salud	"healthcare"	OR	
Salud	"m-health"	OR	"movil health"
Salud	"telemedicine"	AND	
Interacción	HCI	OR	"Interaction"
Interacción	"Usability"	OR	
Interacción	"Interface"	OR	
<b>Cadena de búsqueda general</b>	(IoT OR "Internet of Things" OR "internet-of-things" OR "Internet Of Everything" OR "IoMT" OR "Internet Of Medical Things") AND ("e-health" OR healthcare OR "m-health" OR "electronic health" OR "movil health" OR "telemedicine") AND (HCI OR Interaction OR Usability OR Interface)		

Tabla 3.1: Formación de la cadena de búsqueda para búsquedas automáticas.

### 3.2.4. Período de búsqueda

El período búsqueda está comprendido desde el año 2009 hasta febrero del 2018. La fecha de inicio seleccionada, se debe a que en el año 2009 se acuñó el término IoT para referirse al Internet de las Cosas [6]. Los términos HCI y AAL ya habían sido adoptados con anterioridad a la fecha planteada.

### 3.2.5. Criterios de extracción de datos

Se han definido un conjunto de criterios de extracción de datos los cuales son aplicados a todos los estudios primarios seleccionados para facilitar su clasificación. La Tabla 3.2 lista todos los criterios de extracción de datos identificados para cada sub-pregunta de investigación y las posibles respuestas que estas pueden tener.

COD.	CRITERIO	POSIBLES RESPUESTAS
RQ1: ¿En qué entornos se aplican soluciones basadas en IoT dentro del área de la salud y a quiénes están dirigidas?		
EC1	Entornos	Hospital, Hogar, Áreas de recreación, Vehículo, Laboral, Ciudad, Farmacias y Geriátricos.
EC2	Sector vulnerable	Adultos mayores, Personas con discapacidad, Otros.





EC3	Stakeholders	Pacientes Crónicos, Pacientes Ambulatorios, Cuidadores, Doctores, Otros.
RQ2: ¿Cómo se lleva a cabo la recolección de datos fisiológicos y del entorno de una persona para el uso dentro del área de la salud?		
EC4	Accesorios	Accesorios de ropa, Accesorios para el cuerpo, Accesorios para electrodomésticos, Otros accesorios para el hogar, Accesorios para fuera del hogar, Accesorios para hospitales.
EC5	Equipos de recolección de datos	Monitor médico, Sensores, Smartphone, Kinect, Dispositivo de captación gestual, Eyetracker, Gafas de realidad virtual, Smart Glasses.
EC6	Tipos de sensores y monitores	Sensor térmico, Sensor acústico, Sensor de movimiento, Otros sensores de medio ambiente, Sensor de monitoreo respiratorio, Sensor magnético, Sensor óptico, Sensor para alcohol y cigarro, Sensor de navegación, Sensor de transpiración, Sensor y monitor cardíaco, Sensor de monitoreo hemodinámico, Monitor Neurológico, Monitor Muscular, Monitor Ocular, Sensor de glucosa, Sensor de Visión Artificial, Otros.
RQ3: ¿Qué aplicaciones, implementaciones, soluciones o sistemas propuestos basados en IoT existen para el área de la salud?		
EC7	Sistemas	Context-aware, Ambiente inteligente, Rehabilitación física, Monitoreo de signos vitales, Monitoreo y seguimiento de enfermedades, Control del entorno Monitoreo de medicación, Detección y prevención de caídas Monitoreo del entorno, Monitoreo de actividad humana, Sistema de información hospitalario, Cuidado Pre y Postnatal, Control alimenticio, Detección de obstáculos, Otros.



EC8	Tipo de aplicaciones de software	App, Web, Desktop, Mobile, Sensores Virtuales, Infraestructura Cloud, Base de datos.
EC9	Nivel de implementación	Nivel de datos, Nivel de algoritmos, Nivel GUI.
EC10	Protocolo de comunicación	GPRS, IEEE 802.15.4, ZigBee, XBee, Bluetooth, RFID, WiFi, NFC, 6LoWPAN, USB, HL7, FHIR, GSM, HTTP, IrDA, Otro.
RQ4: ¿Cuáles son las enfermedades, tratamientos y tipos de asistencia más relevantes que pueden ser cubiertos usando soluciones IoT en el área de la salud?		
EC11	Enfermedades	Enfermedades endocrinas, Enfermedades del sistema cardiocirculatorio, Enfermedades del sistema respiratorio, Enfermedades del sistema nervioso, Desórdenes mentales, Lesiones físicas.
EC12	Tratamientos	Emergencia, Discapacidad, Monitoreo y control de enfermedades, Rehabilitación física, Control de medicamentos, Control alimenticio, Otros.
EC13	Asistencia	Asistencia remota, Asistencia personal.
RQ5: ¿Cuáles son las formas de interacción disponibles para el control de soluciones propuestas basadas en IoT en el área de la salud?		
EC14	Interacción	Gestual, Mental, Verbal, Visión, Auditivo, Basadas en sensores y sensibles al contexto, Interfaces WIMP y Touch.
RQ6: ¿De qué manera se ha llevado hasta ahora la investigación en estos temas?		
EC15	Ambiente de creación	Industria, Academia, Ambos.
EC16	Modelo de solución propuesta	Metodología, Proceso, Prototipo, Estudio experimental, Estudio descriptivo.
EC17	Novedad	Nuevo, Extensión.



EC18	Tipo de validación	Prueba de concepto, Caso de estudio, Estudio piloto, Experimento controlado, Quasi-experimento.
EC19	El estudio incluye comité de ética	Si, No.

Tabla 3.2: Criterios de extracción de datos.

### 3.3. Etapa de ejecución

En esta etapa se lleva a cabo el proceso de selección y revisión de los estudios primarios publicados en las revistas digitales y conferencias identificadas en la etapa previa, de esta manera se puede proceder con la extracción de información que responda a las preguntas de investigación planteadas. Las actividades a realizar consisten en la selección de los estudios primarios y el aseguramiento de la calidad.

#### 3.3.1. Selección de estudios primarios

Una vez realizada la búsqueda automática y manual, se procede a realizar la evaluación de cada estudio encontrado para determinar si deben ser incluidos o excluidos del mapeo sistemático. Para esta evaluación se han considerado los siguientes criterios de inclusión:

- Artículos publicados en cualquiera de las fuentes de información especificada para las búsquedas manuales y automáticas.
- Artículos cuya investigación subyacente permita dar respuesta a cualquiera de las sub-preguntas de investigación.

Los criterios de exclusión son los siguientes:

- Artículos introductorios de ediciones especiales: revistas, libros, *workshops*.
- Estudios duplicados en diferentes fuentes.
- Artículos que poseen menos de cinco páginas.
- Artículos que no han sido publicados en inglés.



La selección de estudios primarios se realizó en 3 etapas. La primera etapa consistió en obtener los estudios primarios de las revistas y conferencias definidas, de esta etapa se obtuvieron 996 estudios. En la segunda etapa se llevó a cabo una revisión de los títulos y resúmenes de los estudios primarios obtenidos en la primera etapa, en total se obtuvieron 186 estudios. En la tercera etapa se llevó a cabo la revisión del texto completo de cada estudio primario y se incluyeron los más relevantes del tema, en total se obtuvieron 137 estudios. La Tabla 3.3 muestra un resumen de la cantidad de estudios primarios obtenidos en cada etapa. Los estudios seleccionados se encuentran en el Apéndice A.

BIBLIOTECA DIGITAL	PRIMERA ETAPA	SEGUNDA ETAPA	TERCERA ETAPA	PORCENTAJE
IEEE XPLORE	146	70	60	44 %
ACM	41	12	9	7 %
SpringerLink	788	99	66	48 %
ScienceDirect	21	5	2	1 %
<b>TOTAL</b>	996	186	137	100 %

Tabla 3.3: Número de estudios primarios obtenidos en el proceso de selección.

### 3.3.2. Aseguramiento de la calidad

La calidad de cada estudio se evalúa según el número de citas que se han realizado, para esto se utiliza una escala de Likert de tres puntos. Donde +1 significa que el estudio tiene más de 3 citas, 0 si el estudio tiene entre 1 y 3 citas, y -1 cuando el estudio no cuenta con ninguna cita. La Tabla 3.4 muestra los resultados de la evaluación según los criterios de calificación definidos.

Puntaje	Cantidad	Porcentaje
+1	76	56 %
0	36	26 %
-1	25	18 %
<b>TOTAL</b>	137	100 %

Tabla 3.4: Evaluación de la calidad de los estudios primarios.

## 3.4. Etapa de reporte

Aquí se presentan los resultados obtenidos luego de finalizar la etapa de ejecución. Se han generado tablas en las que se resumen los valores generados



para cada criterio de extracción de manera que, en base a esos resultados, se puedan llevar a cabo comparaciones entre dichos criterios con el fin de descubrir el panorama actual del contexto estudiado dando respuesta a la pregunta de investigación planteada.

### 3.4.1. Porcentajes individuales

Una vez finalizada la etapa de recolección y tabulación de datos, se procede a reportar los resultados alcanzados para cada criterio de extracción. A continuación, se presentan los resúmenes de los valores obtenidos para los criterios de extracción más relevantes, la totalidad de resultados se puede encontrar en el Apéndice B.

Los entornos de aplicación de soluciones IoT en el área de la salud, las personas de los sectores vulnerable a las que van dirigidas y los *stakeholders* se presentan en la Tabla 3.5.

CON	CRITERIO	RESPUESTAS	ESTUDIOS	%
RQ1: ¿En qué entornos se aplican soluciones basadas en IoT dentro del área de la salud y a quiénes están dirigidas?				
EC1	Entornos	Hospital	41	24 %
		Hogar	97	56 %
		Áreas de recreación	17	10 %
		Vehículo	4	2 %
		Laboral	1	1 %
		Ciudad	11	6 %
		Farmacias y geriátricos	2	1 %
EC2	Sector Vulnerable	Adultos mayores	42	66 %
		Personas con discapacidad	18	28 %
		Otros	4	6 %
EC3	Stakeholders	Pacientes Crónicos	25	16 %
		Pacientes Ambulatorios	67	43 %
		Cuidadores	24	15 %
		Doctores	32	21 %
		Otros	8	5 %

Tabla 3.5: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ1.



En la Figura 3.1 se puede observar que la mayor cantidad de estudios se enfocan en el hogar (56 %), seguido de los hospitales (24 %), y áreas de recreación (10 %) como parques, gimnasios, etc. Con menor porcentaje se encuentran los vehículos, la ciudad, el entorno laboral y las farmacias y geriátricos. Estos resultados reflejan que las aplicaciones IoT están destinadas a controlar y monitorear a los pacientes, principalmente, dentro de su hogar, dejando a los hospitales como entornos secundarios.

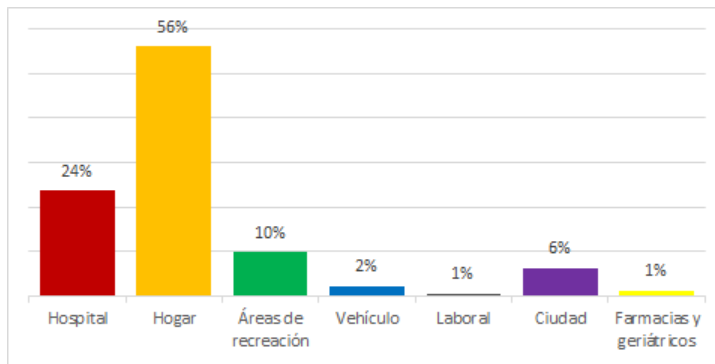


Figura 3.1: EC1 - Porcentaje de entornos para despliegue de soluciones IoT.  
Fuente: Elaboración propia.

Respecto a las partes interesadas, en la Figura 3.2 aparecen, con alto porcentaje, los pacientes ambulatorios (43 %), este tipo de pacientes pueden ser personas con lesiones físicas, estudiantes, bomberos, atletas y personas en general. De forma secundaria, aparecen los pacientes crónicos (16 %) que padecen de enfermedades como diabetes, glucosa hepática, problemas cardiacos, problemas respiratorios, Parkinson, etc. Por otro lado, existen sistemas cuyos interesados son los doctores (21 %) y cuidadores (15 %). En la categoría “Otros” (5 %) se engloban empleados de hospitales, farmacéuticos y laboratoristas.

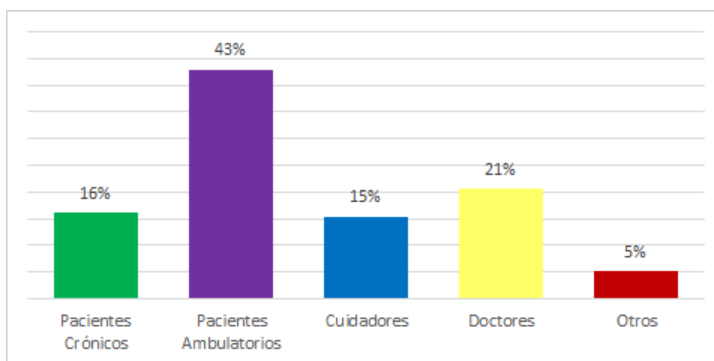


Figura 3.2: EC3 - Porcentaje de partes interesadas a los que van dirigidas las soluciones IoT. Fuente: Elaboración propia.

Los valores obtenidos para las diferentes aplicaciones, implementaciones, soluciones y sistemas IoT en el área de la salud se muestran en la Tabla 3.6.

COD	CRITERIOS	RESPUESTAS	ESTUDIOS	%
RQ3: ¿Qué aplicaciones, implementaciones, soluciones o sistemas propuestos basados en IoT existen para el área de la salud?				
EC7	Sistemas	Context-aware	8	4 %
		Ambiente inteligente	14	8 %
		Rehabilitación física	8	4 %
		Monitoreo de signos vitales	62	34 %
		Control de enfermedades	19	10 %
		Control del entorno	5	3 %
		Monitoreo de medicación	10	5 %
		Detección y prevención de caídas	7	4 %
		Monitoreo del entorno	13	7 %
		Monitoreo de actividad humana	9	5 %



		Sistema de información hospitalario	3	2 %
		Cuidado pre y postnatal	2	1 %
		Control alimenticio	3	2 %
		Detección de obstáculos	3	2 %
		Otros sistemas	19	10 %
EC8	Tipo de Aplicación de Software	Aplicaciones móviles	73	27 %
		Aplicaciones Web	48	18 %
		Aplicaciones Desktop	27	10 %
		Sensores/ Mobile	18	7 %
		Sensores Virtuales	1	0 %
		Infraestructura cloud	44	16 %
		Base de datos	59	22 %

Tabla 3.6: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ3.

En la Figura 3.3 se presentan los sistemas propuestos basados en IoT para el área de la salud. Las propuestas con mayor enfoque, son los sistemas encargados del monitoreo de los signos vitales (34 %) como frecuencia cardíaca, respiración, etc. En menor medida se proponen sistemas para monitoreo y control de enfermedades (10 %) sea de forma personal o remota y estudios que se enfocan en crear ambientes inteligentes, adaptativos y auto-configurables para el confort de las personas (8 %). Finalmente, existen varios sistemas propuestos que gozan de un menor impacto como son los sistemas para control de medicación (5 %), monitoreo de actividad humana (5 %), entre otros. En la categoría “Otros”, se engloban sistemas como monitoreo del sueño, recordatorio automático de citas, monitoreo del estado de ánimo, control de caja de medicinas inteligente, etc.



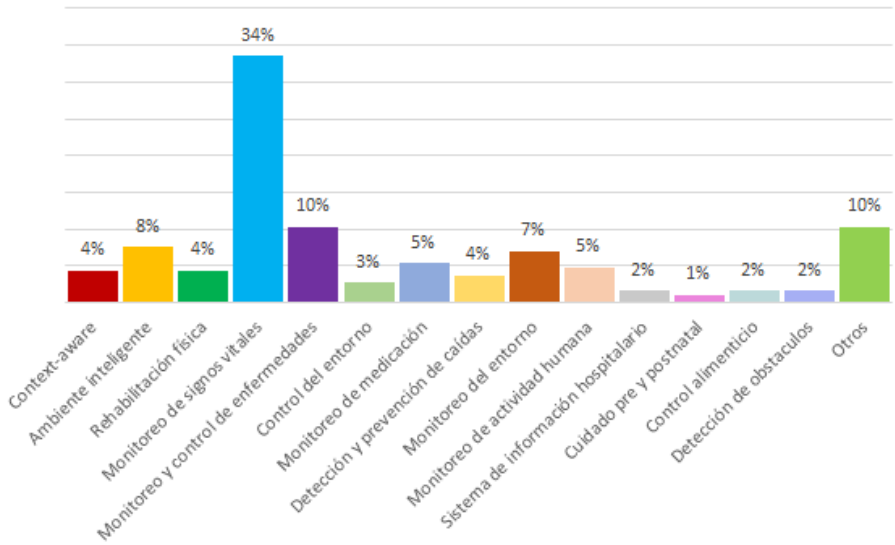


Figura 3.3: EC7 - Porcentaje de tipos de soluciones IoT propuestas. Fuente: Elaboración propia.

Los resultados obtenidos para los tipos de enfermedades y tratamientos que se pueden cubrir usando soluciones IoT en el área de la salud se muestran en la Tabla 3.7.

COD	CRITERIOS	RESPUESTAS	ESTUDIOS	%
RQ4: ¿Cuáles son las enfermedades, tratamientos y tipos de asistencia más relevantes que pueden ser cubiertos usando soluciones IoT en el área de la salud?				
EC11	Enfermedades	Enfermedades endocrinas	7	27 %
		Enfermedades del sistema cardiocirculatorio	6	23 %
		Enfermedades del sistema respiratorio	2	8 %
		Enfermedades del sistema nervioso	4	15 %



		Desórdenes mentales	3	12 %
		Lesiones físicas	4	15 %
EC12	Tratamientos	Emergencia	7	5 %
		Discapacidad	8	6 %
		Monitoreo y control de enfermedades	91	69 %
		Rehabilitación física	9	7 %
		Control de medicamentos	10	8 %
		Control alimenticio	5	4 %
		Otros	1	1 %
EC13	Asistencia	Asistencia remota	92	78 %
		Asistencia personal	26	22 %

Tabla 3.7: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ4.

Las enfermedades en las que mayor énfasis se pone son las enfermedades endocrinas (27 %) y enfermedades del sistema cardiocirculatorio (23 %). El monitoreo y control de enfermedades aparece como tratamiento principal (69 %). Finalmente, la mayoría de sistemas propuestos son concebidos con la finalidad de brindar asistencia remota (78 %).

Los diferentes tipos de interacción propuestas para el control de soluciones IoT en el área de la salud y los valores obtenidos para cada uno se muestra en la Tabla 3.8.



COD	CRITERIOS	RESPUESTAS	ESTUDIOS	%
RQ5: ¿Cuáles son las formas de interacción disponibles para el control de soluciones propuestas basadas en IoT en el área de la salud?				
EC14	Interacción	Gestual	9	4 %
		Mental	3	1 %
		Verbal	8	4 %
		Visión	4	2 %
		Auditivo	8	4 %
		Basadas en sensores y sensibles al contexto	95	44 %
		Interfaces WIMP y Touch	87	41 %

Tabla 3.8: Porcentajes individuales para el criterio de extracción de la subpregunta RQ5.

En la Figura 3.4 se puede observar que los estudios primarios se enfocan, principalmente, en interacciones basadas en sensores y sensibles al contexto (44 %) e interacciones con interfaces WIMP y Touch (41 %) mediante interfaces gráficas desplegadas ya en computadoras, tablets o smartphones. Con un porcentaje inferior se encuentran la interacción gestual (4 %), interacción mediante comandos de voz (4 %) y la interacción auditiva (4 %) la cual se basa en emitir sonidos que alertan a los pacientes o usuarios sobre amenazas, alarmas o recordatorios. Por último, las interacciones con menor énfasis son aquellas que se realizan mediante la visión (2 %) y las que usan interfaces cerebrales para la ejecución de comandos (1 %).

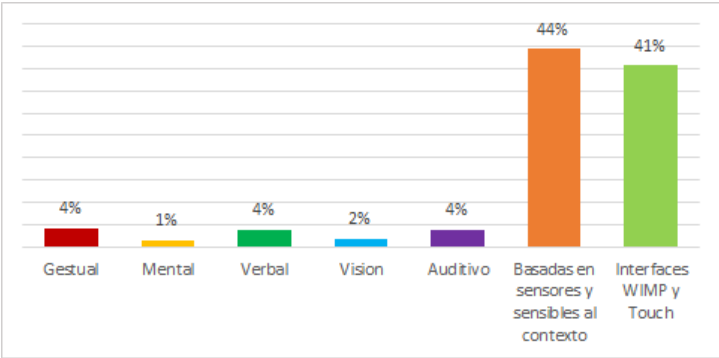


Figura 3.4: EC14 - Porcentaje de tipos de interacción propuestas para controlar soluciones IoT. Fuente: Elaboración propia.



### 3.4.2. Comparación de criterios

Con los resultados obtenidos del proceso de revisión de estudios publicados que responden a la pregunta de investigación planteada, se pueden establecer comparaciones y contrastar diferentes criterios de extracción con el fin de obtener el estado del arte en lo que respecta a los sistemas, aplicaciones y soluciones basadas en IoT para el área de la salud.

La Figura 3.5 muestra una comparación entre los criterios EC2: Sector Vulnerable (horizontal izquierda) y EC1: Entornos (horizontal derecha) con EC14: Interacción (vertical). Respecto al sector vulnerable, se puede observar que la interacción que proponen la mayoría de estudios primarios para con los adultos mayores está basada en sensores, esto debido a que el monitoreo de sus signos vitales y actividades cotidianas es su enfoque principal, lo mismo sucede con la interacción ofrecida para las personas con discapacidad. Por otro lado, muchos de los sistemas propuestos ofrecen también aplicaciones web, de escritorio y apps para recopilación de datos y presentación de resultados, aquí el usuario interactúa con interfaces WIMP y *Touch*. Las interacciones basadas en gestos, interfaces cerebrales, comandos de voz, visión y auditivas son exploradas en menor medida.

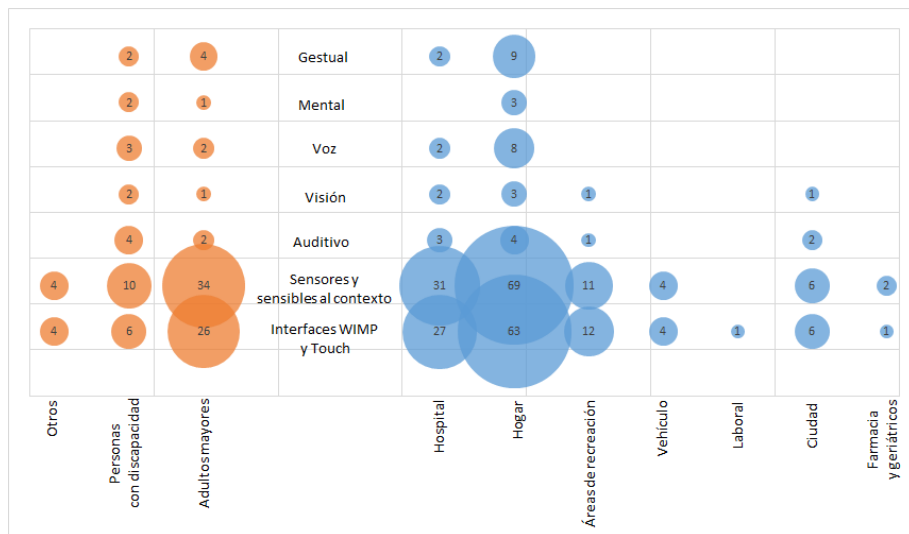


Figura 3.5: Comparación de EC2: Sector Vulnerable y EC1: Entornos entre EC14: Interacción. Fuente: Elaboración propia.

Considerando los entornos de despliegue de los sistemas propuestos, en el

hogar priman las interacciones basadas en sensores e interfaces WIMP y *Touch*, en menor medida, aparecen las interacciones en entornos hospitalarios y las áreas de recreación (parques, gimnasios, etc.). Es notable también el uso de interacciones basadas en gestos y comandos de voz para el control del entorno dentro del hogar.

En la Figura 3.6 se realiza una comparación entre el criterio EC7: Sistemas (horizontal) y EC14: Interacción (vertical). Las interacciones gestuales y por comandos de voz son usada principalmente en sistemas para inteligencia ambiental, monitoreo de signos vitales y control de enfermedades. Las interacciones auditivas son más usadas en sistemas de detección de obstáculos debido a que emiten sonidos que alertan a las personas de la presencia de objetos con los que podrían colisionar. La interacción mental se usa en sistemas para control del entorno y las interacciones visuales para sistemas de rehabilitación física. Finalmente, las interacciones basadas en sensores e interacciones con interfaces WIMP y *Touch* se usan en casi todos los sistemas propuestos estando mayormente presentes en sistemas para monitoreo de signos vitales y control de enfermedades.

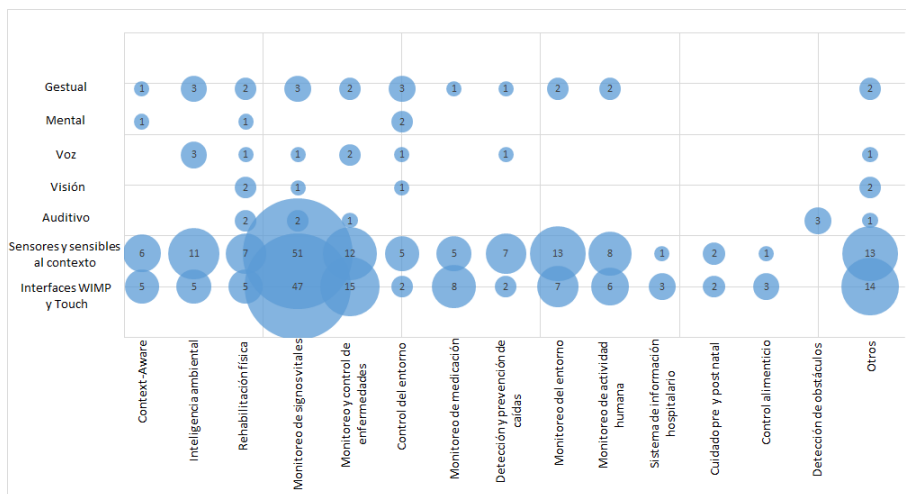


Figura 3.6: Comparación de EC7: Sistemas y EC14: Interacción. Fuente: Elaboración propia.

La Figura 3.7 muestra una comparación entre el criterio de extracción EC2: Sector vulnerable (vertical) con los criterios EC12: Tratamientos (horizontal izquierda) y EC4: Accesorios (horizontal derecha). Para los adultos mayores se observa que los tratamientos son: monitoreo y control de enfermedades,



control de medicamentos. Mientras que para las personas con discapacidad el tratamiento principal es la rehabilitación física. Para otros sectores vulnerables (mujeres embarazadas y niños) el tratamiento principal es el monitoreo y control de enfermedades. En cuanto a los accesorios que se emplean con respecto a cada sector vulnerable se tiene que para los adultos mayores se utilizan accesorios para el cuerpo. Las personas con discapacidad hacen uso de accesorios para el hogar capaces de detectar cualquier cambio en el ambiente.

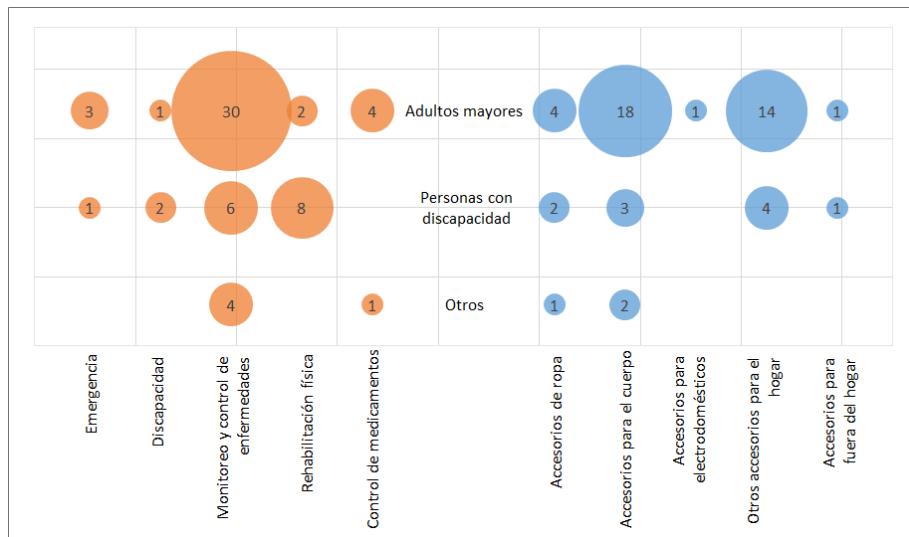


Figura 3.7: Comparación de EC12: Tratamientos y EC4: Accesorios entre EC2: Sector Vulnerable. Fuente: Elaboración propia.

La Figura 3.8 muestra una comparación entre el criterio de extracción EC16: Modelo de solución propuesta (vertical) con los criterios EC15: Ambiente de creación (horizontal izquierda) y EC4: Tipo de validación (horizontal derecha). El prototipo es la solución más empleada y que se utiliza principalmente en ambientes académicos. Los estudios descriptivos se han desarrollado en un ambiente académico. Las validaciones que generalmente se llevan a cabo para los prototipos se desarrollan mediante pruebas de conceptos y experimentos controlados. Los estudios descriptivos se validan mediante pruebas de conceptos.

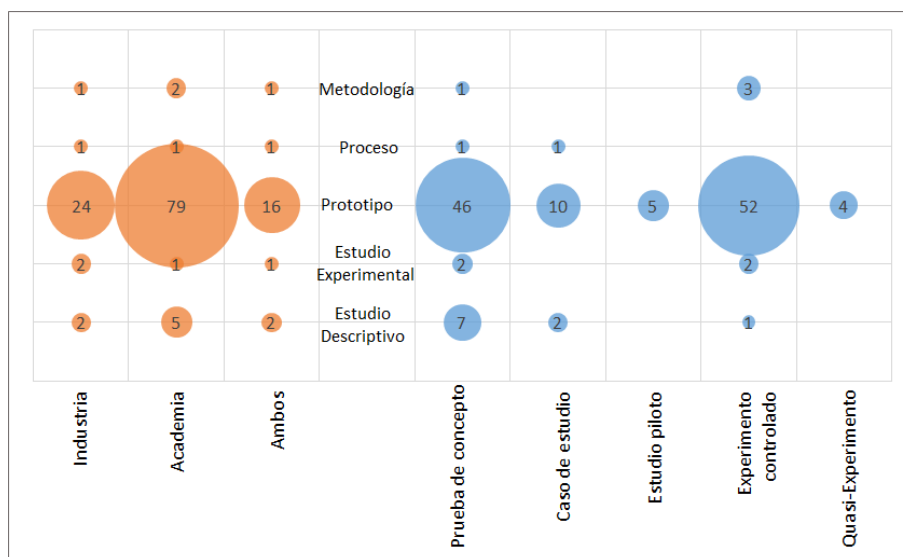


Figura 3.8: Comparación de EC15: Ambiente de creación y EC18: Tipo de validación entre EC16: Modelo de solución propuesta. Fuente: Elaboración propia.

### 3.4.3. Discusión

Esta sub-sección presenta una síntesis de los resultados obtenidos luego de llevar a cabo la revisión de los estudios primarios. La Figura 3.9 muestra el número de estudios incluidos en el mapeo sistemático de cada año dentro del periodo de búsqueda desde el 2009 hasta 2018. Se puede observar que los años 2016, 2017, 2015 y 2013 son en los que se han publicado la mayor cantidad de estudios referentes al contexto estudiado, esto demuestra el gran interés que ha tenido la comunidad científica en los últimos años por brindar soluciones basadas en IoT para el cuidado de la salud de las personas.

A continuación, en base a los resultados obtenidos, se da a conocer las respuestas a las diferentes sub-preguntas de investigación planteadas en este capítulo:

*RQ1: ¿En qué entornos se aplican las soluciones basadas en IoT dentro del área de la salud y a quiénes están dirigidas?*

Respecto a los entornos de aplicación de las soluciones basadas en IoT dentro del área de la salud, se ha encontrado que los hogares son la principal consideración al momento de desplegar los diferentes sistemas propuestos, aquí se puede encontrar sistemas que permiten el monitoreo de pacientes diabéticos,

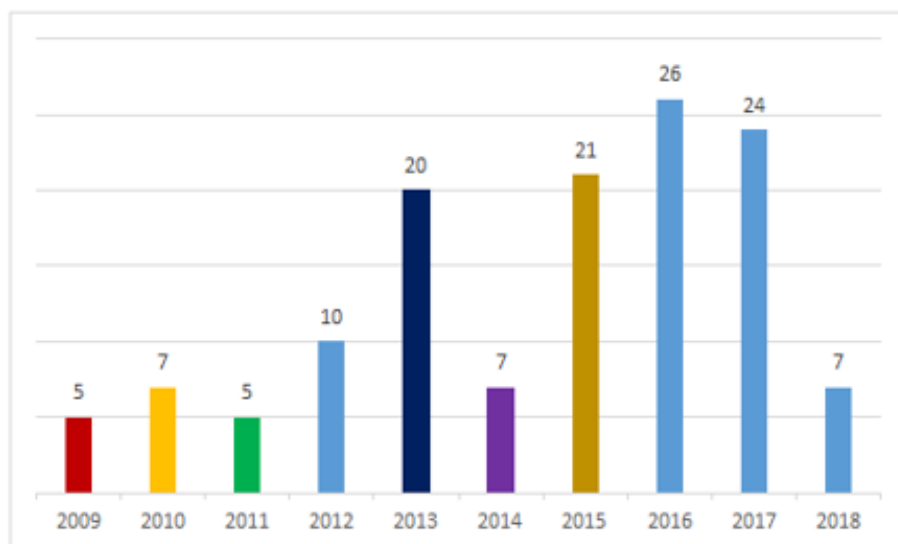


Figura 3.9: Cantidad de estudios por año seleccionados para el mapeo sistemático. Fuente: Elaboración propia.

asistentes de rehabilitación física, monitoreo de enfermos con Parkinson, control y monitoreo del entorno, entre otras. Los entornos hospitalarios aparecen como opción secundaria, aquí se despliegan sistemas de información hospitalarios, sistemas para el control de medicación, detección de alergias, entre otros. Las soluciones propuestas se enfocan ampliamente en el cuidado tanto de adultos mayores como de personas con discapacidad. Entre las partes interesadas con mayor consideración están los enfermos ambulatorios, crónicos, doctores y cuidadores.

*RQ2: ¿Cómo se lleva a cabo la recolección de datos fisiológicos y del entorno de una persona para el uso dentro del área de la salud?*

Considerando las formas de recolección de datos de los pacientes, usuarios y entornos a los que van dirigidas las diferentes soluciones basadas en IoT, se ha encontrado que se usan frecuentemente accesorios para el cuerpo, la ropa y el hogar, ya sea para monitoreo de signos vitales, monitoreo de actividad física o control del entorno. Para la captura de datos se tiende a usar sensores, monitores médicos y *smartphones*. En cuanto a los tipos de sensores, se ha determinado que los más empleados son los que permiten recopilar información cardíaca, información térmica (corporal y del entorno), información de movi-





miento, presión sanguínea (monitoreo hemodinámico) y monitoreo del sistema respiratorio.

*RQ3: ¿Qué aplicaciones, implementaciones, soluciones o sistemas propuestos basados en IoT existen para el área de la salud?*

En lo que se refiere a los tipos de sistemas propuestos, se ha encontrado que la mayoría se enfoca en el monitoreo de los signos vitales de los usuarios o pacientes, control de enfermedades ambulatorias y crónicas, creación de ambientes inteligentes, control de medicación, monitoreo y control de entornos. Entre los tipos de aplicaciones de software que se han creado predominan las aplicaciones móviles y aplicaciones Web; además, varios sistemas han sido creados usando infraestructuras *Cloud*, servidores y bases de datos. Los protocolos de comunicación y transmisión de datos entre sensores y aplicaciones que más se han empleado son *WiFi*, *Bluetooth* y *ZigBee*.

*RQ4: ¿Cuáles son las enfermedades, tratamientos y tipos de asistencia más relevantes que pueden ser cubiertos usando soluciones IoT en el área de la salud?*

Las enfermedades que generalmente se cubren por las soluciones propuestas son tanto las enfermedades endocrinas como las enfermedades del sistema cardiocirculatorio y respiratorio. Los tratamientos se enfocan, en su mayoría, en proveer monitoreo de información vital y control a dichas enfermedades. Además, predomina la asistencia remota por sobre la asistencia personal.

*RQ5: ¿Cuáles son las formas de interacción disponibles para el control de soluciones propuestas basadas en IoT en el área de la salud?*

Respecto a la interacción que ofrecen a los usuarios y pacientes las diferentes soluciones basadas en IoT, se ha encontrado que predominan aquellas interacciones basadas en sensores y sensibles al contexto, junto con las interacciones a través de interfaces *WIMP* y *Touch*, las cuales, en la actualidad, son las más comunes debido a la gran cantidad de aplicaciones móviles, Web y de escritorio que han ido apareciendo. Existe un campo de investigación bastante amplio que debe ser considerado para hacer que las personas puedan interactuar con el entorno usando interacciones gestuales, mentales, verbales y a través de la visión ya que cada una de ellas se puede acoplar a las necesidades de diferentes tipos de usuarios, por ejemplo, personas con ceguera pueden interactuar a través de gestos, personas con movilidad reducida pueden emplear interacción mental o comandos de voz, etc., y no obligarlas a interactuar con interfaces gráficas que, por lo general, no están pensadas para personas que carecen de ciertas facultades lo cual dificulta el uso de tales sistemas.



*RQ6: ¿De qué manera se ha llevado hasta ahora la investigación en estos temas?*

Considerando la forma en la que se ha llevado a cabo los diferentes estudios primarios, la academia es la que mayor investigación realiza por sobre la industria. Los modelos de solución que se proponen son, en su mayoría, prototipos que se validan usando pruebas de conceptos o experimentos controlados.

En base a los resultados obtenidos para cada sub-pregunta de investigación, se puede establecer la respuesta a la pregunta de investigación general:

*RQ: ¿Cuáles son y bajo qué entornos funcionan las soluciones basadas en IoT dentro del dominio de la salud, a quiénes están dirigidas, cómo se recolecta la información para los diferentes tratamientos que cubren y que tipos de interacción ofrecen?*

Se ha encontrado que los hogares y los hospitales se consideran como entornos primordiales para el despliegue de soluciones IoT en el dominio de la salud, estas soluciones se enfocan ampliamente en el cuidado tanto de adultos mayores como de personas con discapacidad. Para la recolección de información sanitaria, frecuentemente se usan accesorios y/o dispositivos para el cuerpo, la ropa y el hogar, ya sea para monitoreo de signos vitales, monitoreo de actividad física o control del entorno; los datos son capturados usando sensores, monitores médicos y *smartphones*, estos dispositivos permiten recopilar información cardíaca, información térmica, información de movimiento, presión sanguínea entre otros. En cuanto a las soluciones propuestas, la mayoría se enfoca en el monitoreo de los signos vitales de los usuarios o pacientes, control de enfermedades ambulatorias y crónicas, creación de ambientes inteligentes, control de medicación, monitoreo y control de entornos. Respecto a los tipos de interacción que ofrecen, predominan aquellas basadas en sensores y sensibles al contexto, junto con las interacciones a través de interfaces WIMP y *Touch*, en menor medida aparecen las interacciones naturales como son las interacciones gestuales, mentales, verbales y a través de la visión.

### 3.5. Conclusiones

En este capítulo se ha presentado un mapeo sistemático basado en una revisión de la literatura cuya finalidad fue la de explorar el estado actual de las investigaciones en el contexto de las soluciones (sistemas, aplicaciones, implementaciones) basadas en IoT para el cuidado de la salud y la forma en la que se lleva a cabo la interacción entre las personas y dichas soluciones.

Para realizar el mapeo sistemático, se adoptó la metodología propuesta por



Kitchenham y Charters [45], la cual sugiere la realización de un conjunto de actividades específicas que se dividen en tres etapas: i) Planificación, en donde se definieron las preguntas de investigación, estrategias de búsqueda, amenazas a la validez, período y cadena de búsqueda, criterios de extracción de datos, entre otros; ii) Ejecución, aquí se llevó a cabo el proceso de selección y revisión de los estudios primarios publicados en las revistas digitales y conferencias identificadas en la etapa anterior, para proceder con la extracción de información que responda a las preguntas de investigación planteadas; iii) Reporte, en donde se presentaron los resultados obtenidos luego de finalizar la etapa de ejecución.

Los resultados obtenidos en el mapeo sistemático muestran el interés que existe en el campo de la investigación por crear soluciones que permitan brindar y fomentar el cuidado de la salud de sectores vulnerables en donde se consideran, principalmente, a los adultos mayores y personas con discapacidad. El monitoreo de signos vitales, control de enfermedades, control y monitoreo del entorno y la inteligencia ambiental son puntos mayormente abordados en los diferentes estudios primarios, esto da paso al despliegue de sistemas en entornos de vida asistidos para el cuidado y alargamiento de la vida de adultos mayores y personas con discapacidad. Sin embargo, existe una brecha bastante marcada en la interacción que se ofrece, esto debido a que son muy pocos los estudios primarios que se enfocan en brindar soluciones que permitan a las personas interactuar de forma más natural con los sistemas propuestos, ya sea mediante interacción mental, visual, comandos de voz, etc., de manera que tales sistemas sean más accesibles a personas que no poseen todas las características necesarias para interactuar con sistemas que ofrecen únicamente interfaces gráficas.





## Capítulo 4

# Arquitectura de software: AALIIoT

En este capítulo se presenta la arquitectura de software propuesta basada en las Interacciones Humano-Computador para aplicaciones de Internet de las Cosas (IoT) en Ambientes de Vida Asistidos (AAL). El capítulo se distribuye de la siguiente manera: la sección 4.1 presenta la descripción general de la arquitectura; la sección 4.2 presenta la arquitectura desde un punto de vista estático, este punto de vista muestra, de manera general, los componentes que tiene la arquitectura; la sección 4.3 muestra la arquitectura desde un punto de vista funcional, describiendo el funcionamiento de cada componente; la sección 4.4 muestra la arquitectura desde un punto de vista dinámico, explicando cómo se comportan los componentes a lo largo del tiempo y cómo éstos interactúan.



## 4.1. Descripción general

La arquitectura de software propuesta recibe el nombre de AALIIoT, el cual hace referencia a las tres áreas de dominio de las cuales forma parte. El nombre se forma considerando los siguientes acrónimos: “AAL”, tomado de *Ambient Assited Living*; “I”, tomada de *Human-Computer Interactions*; “IoT”, tomado de *Internet of Things*.

La arquitectura de software AALIIoT, está diseñada para ser el soporte de soluciones basadas en IoT, las mismas que se despliegan en entornos de vida asistida (hogar, geriátricos, hospitales, etc.). Tales soluciones usan diferentes tipos de dispositivos o sensores conectados a internet para brindar asistencia a grupos de usuarios vulnerables: adultos mayores y personas con discapacidad. La arquitectura contempla la posibilidad de que los dispositivos, sensores o actuadores sean controlados con o sin intervención del usuario:

- Control sin intervención del usuario: hace referencia a la situación en la que los dispositivos o sensores ejecutan cierta acción en base a la interpretación de datos recolectados. Por ejemplo, un dispositivo que mide la temperatura y la regula automáticamente en base a las necesidades o preferencias de una persona.
- Control con intervención del usuario: se refiere a la situación en la que los dispositivos o sensores necesitan una orden del usuario para ejecutar una acción, dicha orden es recogida por un dispositivo que actúa como receptor de comandos de interacción. Por ejemplo, abrir una puerta cuando una persona hace un gesto específico con una mano.

AALIIoT considera los tipos de interacción natural más frecuentes en ambientes de vida asistidos: i) interacción vocal, ii) interacción visual, iii) interacción gestual, iv) interacción mental, vi) interacción basada en sensores, vii) interacción con interfaces gráficas de usuario y los integra con los diferentes dispositivos, sensores y actuadores que forman parte de las aplicaciones encargadas de brindar una mejor calidad de vida a sus usuarios. La arquitectura de software AALIIoT se presenta en la Figura 4.1.

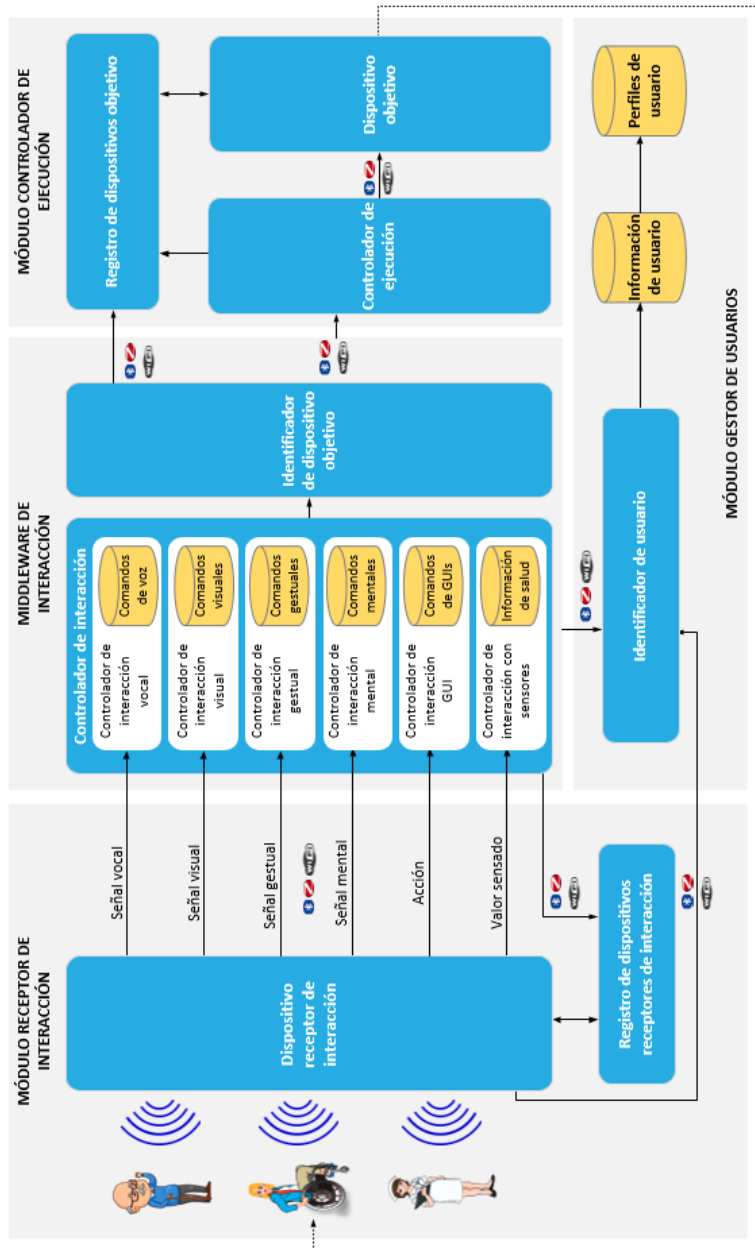


Figura 4.1: Arquitectura de software AALIIoT. Fuente: Elaboración propia.



Las siguientes secciones brindan una explicación más detallada de las diferentes partes que componen la arquitectura, su funcionamiento y comportamiento a lo largo del tiempo.

## 4.2. Vista estática

AALIIoT puede ser vista como la unión de cuatro módulos: i) Módulo Gestor de Usuarios, ii) Módulo Receptor de Interacción, iii) Middleware de Interacción y iv) Módulo Controlador de Ejecución.

### 4.2.1. Módulo Gestor de Usuarios

El Módulo Gestor de Usuarios (Figura 4.2) es el encargado de identificar al usuario dentro del entorno y habilitar la o las interacciones adecuadas para él en base a un conjunto de datos personales almacenados en un repositorio de información de usuarios y un repositorio de perfiles de usuarios.



Figura 4.2: Módulo Gestor de Usuarios. Fuente: Elaboración propia.

Tres componentes forman parte de este módulo; a continuación, se describe cada uno de ellos:

- **Identificador de usuario:** componente encargado de identificar o reconocer al usuario dentro del entorno.
- **Repositorio de información de usuario:** almacena la información personal de un usuario y los datos necesarios para validar la autenticación.
- **Repositorio de perfiles de usuario:** almacena los diferentes perfiles de usuario que serán asignados a las personas que usarán el sistema dentro del entorno, de manera que cada usuario podrá interactuar con el sistema empleando las interacciones definidas en su perfil asignado.



## 4.2.2. Módulo Receptor de Interacción

El Módulo Receptor de Interacción (Figura 4.3) se encarga de reconocer los comandos (implícitos o explícitos) de los usuarios mediante dispositivos receptores de interacción o sensores para su posterior ejecución.

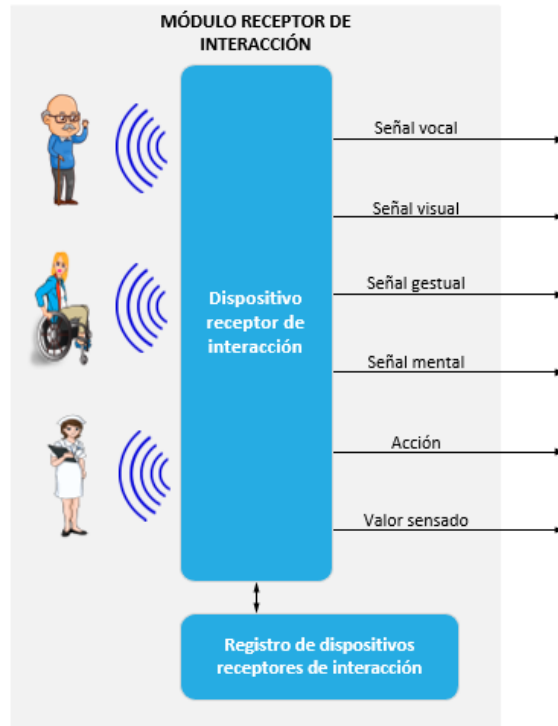


Figura 4.3: Módulo Receptor de Interacción. Fuente: Elaboración propia.

Existen dos componentes que forman parte de este módulo los cuales se describen a continuación:

- **Dispositivo receptor de interacción:** Son todos los dispositivos capaces de captar las interacciones del usuario con el sistema ya sea mediante gestos, señales auditivas, movimiento de los ojos, comandos mentales, acciones realizadas en dispositivos con GUIs, o valores medidos por sensores. La información recogida representa la solicitud realizada por el usuario para llevar a cabo una acción específica. Entre los dispositivos receptores de interacción, podemos encontrar los siguientes: micrófono,

videocámara, casco receptor de señales EEG, sensor de temperatura, entre otros.

- **Registro de dispositivos receptores de interacción:** Se refiere a la especificación y detalle de las propiedades de cada uno de los dispositivos receptores de interacción disponibles en el entorno.

### 4.2.3. Middleware de Interacción

El Middleware de Interacción (Figura 4.4) se encarga de interpretar las señales enviadas por el Módulo Receptor de Interacción y definir el dispositivo o dispositivos a los cuales van dirigidas para que ejecuten la acción solicitada por el usuario.

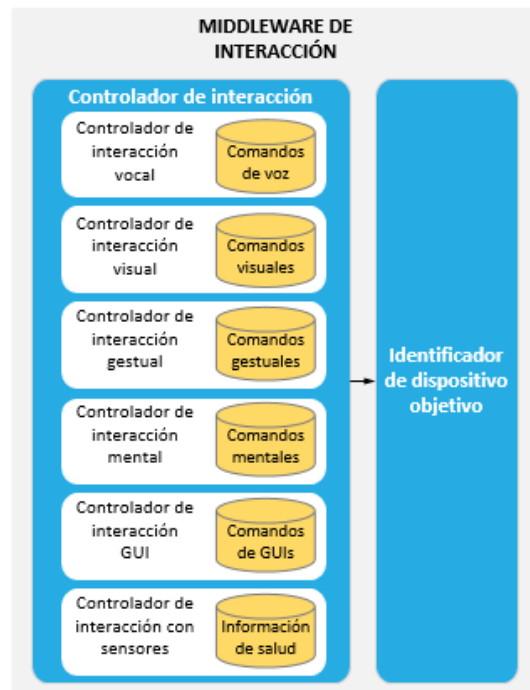


Figura 4.4: Middleware de Interacción. Fuente: Elaboración propia.

El Middleware de Interacción está compuesto de las partes que se describen a continuación:

- **Controlador de interacción:** Procesa e interpreta los comandos recibidos basados en algoritmos específicos y en repositorios de comandos disponibles para cada forma de interacción. Se divide en los siguientes subcomponentes:
  - Controladores de interacción vocal, visual, gestual, mental, basada en sensores y GUI.
  - Repositorio de comandos de voz, visuales, gestuales, mentales.
  - Repositorio de información de salud.
- **Identificador de dispositivo objetivo:** Componente que identifica el dispositivo final al que va dirigido el comando enviado por el usuario.

#### 4.2.4. Módulo Controlador de Ejecución

En el Módulo Controlador de Ejecución (Figura 4.5) se ejecutan los comandos enviados por el usuario. Cada dispositivo o actuador cambiará de estado según la solicitud del usuario.

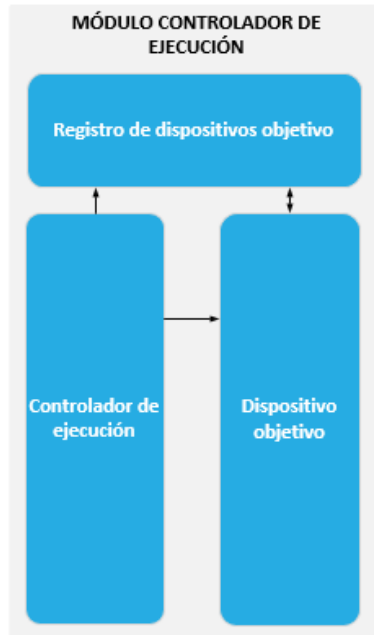


Figura 4.5: Módulo Controlador de Ejecución. Fuente: Elaboración propia.



El módulo se conforma de los siguientes componentes:

- **Controlador de ejecución:** Prepara y envía la acción a ejecutar al dispositivo o actuador objetivo.
- **Registro de dispositivos objetivo:** Se refiere a la especificación y detalle de las propiedades, dirección en la red y comandos que aceptan cada uno de los dispositivos o actuadores presentes en el entorno.
- **Dispositivo objetivo:** Representa a los diferentes dispositivos que pueden ser controlados dentro del entorno, por ejemplo: bombilla de luz, puerta eléctrica, persiana eléctrica, ventana eléctrica, calefactor, etc.

## 4.3. Vista funcional

El propósito de esta sección es describir los elementos funcionales del sistema, sus responsabilidades principales, interfaces e interacciones. A continuación, se describe cada componente en función de sus responsabilidades, dando a conocer sus interfaces e interacciones con otros componentes.

### 4.3.1. Componentes del Módulo Gestor de Usuarios

#### 4.3.1.1. Identificador de usuarios

##### Responsabilidades

Se encarga de realizar la autenticación del usuario dentro del entorno. Esta autenticación es necesaria para que el sistema o aplicación tenga conocimiento de los tipos de interacción que debe habilitar para dicho usuario y, en consecuencia, los comandos que éste podrá enviar. Un usuario puede interactuar ya sea usando solamente un tipo de interacción o, inclusive, todos los tipos de interacción disponibles en base a la información especificada en el Repositorio de perfiles de usuario o, a través de algoritmos avanzados que detectan los problemas físicos y de salud de la persona que hará uso del sistema o aplicación y habilite la interacción necesaria.

##### Interacciones

- **Repositorio de información de usuario:** A través de este repositorio, el Identificador de usuarios puede comparar y recuperar la información de autenticación.



- **Repositorio de perfiles de usuario:** Mediante este repositorio, el Identificador de usuarios puede recuperar las necesidades de interacción disponibles para un usuario específico y habilitarlas para su uso en un entorno determinado.
- **Dispositivo receptor de interacción:** Para llevar a cabo una autenticación, el usuario debe interactuar con algún dispositivo, este dispositivo capturará las señales y/o patrones necesarios (rasgos faciales, iris, voz, etc.) y las enviará al Identificador de usuarios para que lleve a cabo el proceso de autenticación.
- **Middleware de Interacción:** El Identificador de usuarios envía la información de los tipos de interacción que el usuario necesita, en base a su perfil de usuario, al Middleware de Interacción para que sean habilitadas.

## Interfaces

Este componente expone una interfaz de comunicación para que el Middleware de Interacción pueda acceder a consultar información como:

- La autenticación de un usuario.
- Su perfil de usuario en donde se especifican el tipo de interacciones que deben ser habilitadas para dicho usuario.

## Técnicas de implementación

Este componente puede ser implementado usando diferentes técnicas de autenticación o identificación de usuarios. Pueden usarse las técnicas tradicionales como el uso de una contraseña basada en caracteres, el uso de tarjetas de identificación, o bien, pueden emplearse técnicas más avanzadas y precisas basadas en el uso de información biométrica. La biometría es un sistema de reconocimiento de patrones utilizado para autenticar o identificar a una persona a través de sus propias características fisiológicas o de comportamiento [90]. Tales características pueden incluir huellas dactilares, iris, rasgos faciales, ondas cerebrales, geometría de la mano, reconocimiento de la voz, entre otros.

Es posible que algunos sistemas o aplicaciones requieran realizar la identificación siempre que el usuario solicita la ejecución de un comando, esta situación genera un coste computacional elevado, debido a la necesidad de ejecutar algoritmos de autenticación y algoritmos de control de interacción al mismo tiempo. Por otro lado, existirán sistemas que requieran de una única identificación, o bien, habrá sistemas que no requiera de ningún tipo de identificación de usuarios, por lo cual este componente puede ser deshabilitado si no resulta necesario.



#### 4.3.1.2. Repositorio de información de usuario

##### Responsabilidades

Se encarga de almacenar información de los usuarios, la misma que es necesaria para el sistema o aplicación. La información puede variar dependiendo de los requerimientos bajo los cuales fueron diseñados tales sistemas, por ejemplo, existirán sistemas que requieran los nombres de la persona, información de los cuidadores, direcciones, etc., mientras que otros sistemas tal vez no consideren necesaria dicha información y se centren en otro tipo de datos.

La información que sí debe ser considerada es aquella que será necesaria para la autenticación del usuario, por ejemplo, si la autenticación será a través de la verificación de los rasgos faciales, entonces debe existir un sub-repositorio en donde se almacene la información de los rasgos faciales de una persona para su comprobación durante el proceso de identificación.

##### Interacciones

- **Identificador de usuarios:** Accede al repositorio para recuperar los datos con los cuales realizar el reconocimiento de personas.
- **Repositorio de perfiles de usuario:** Un registro en el Repositorio de información de usuario puede almacenar referencias a una o varias instancias del Repositorio de perfiles de usuario para conocer las necesidades de interacción de una persona determinada.

##### Interfaces

La interfaz que expone este componente depende del tipo de implementación (archivos planos, bases de datos, etc.). Entre los datos de salida de este componente se encuentran los siguientes:

- Información general del usuario.
- Información necesaria para la identificación del usuario.

##### Técnicas de implementación

Este componente puede ser implementado usando archivos de texto plano, bases de datos SQL, bases de datos NoSQL o cualquier forma de almacenamiento. Además, es recomendable que esté formado por sub-repositorios relacionados entre sí que almacenen tipos de información específica, por ejemplo,



un sub-repositorio que guarda la información general del usuario y otros sub-repositorios que guardan la información necesaria para la identificación: contraseñas, parámetros de huellas digitales, información de retina, rasgos faciales, etc.

#### 4.3.1.3. Repositorio de perfiles de usuario

##### Responsabilidades

Se encarga de almacenar información relacionada a los perfiles que un usuario puede tener dentro del entorno para habilitar las interacciones definidas en dicho perfil mediante las cuales el usuario puede controlar los dispositivos presentes. Un perfil de usuario es una representación digital de los datos únicos que pertenecen a un usuario en particular; además, el modelado del perfil de usuario implica la creación de una estructura de datos que puede contener los atributos característicos de un tipo de usuarios [75]. Esto se puede hacer definiendo manualmente estos atributos, p. hábitos, preferencias e intereses o inferir información no observable a partir de datos observables relacionados con sus acciones, pensamientos o comportamientos [91].

En este trabajo, los perfiles de usuario permiten definir características ya sea para un usuario específico o para un grupo de usuarios con necesidades similares, por ejemplo, en una habitación pueden existir personas que, debido a sus limitaciones físicas, son capaces de encender una bombilla de luz realizando gestos con las manos; por otro lado, en una segunda habitación conviven personas que pueden encender una bombilla de luz usando, únicamente, comandos vocales. En este ejemplo, existen dos perfiles de usuario, cada usuario dentro de una misma habitación tendrá un mismo perfil. Sin embargo, estos perfiles pueden funcionar de la misma manera para usuarios con diferentes necesidades de interacción que conviven en el mismo entorno controlando un mismo dispositivo, para ello será necesaria la identificación del usuario y el perfil asignado, siempre que éste envíe un comando para poder diferenciarlo de los demás.

##### Interacciones

- **Identificador de usuarios:** Este repositorio mantiene disponibles los perfiles de usuario definidos para que el identificador de usuario pueda acceder a ellos cuando sea necesario.
- **Repositorio de perfiles de usuario:** Uno o varios registros en el Repositorio de perfiles de usuario pueden ser referenciados por una o varias instancias del Repositorio de información de usuario para conocer las necesidades de interacción de una persona determinada.



## Interfaces

La interfaz que expone este componente depende del tipo de implementación (archivos planos, bases de datos, etc.). Este componente expone toda la información concerniente al perfil de usuario, como, por ejemplo, la descripción del perfil, el entorno en el que actúa, los tipos de interacción a los que el usuario puede acceder, los comandos que puede ejecutar, etc.

## Técnicas de implementación

Este componente puede ser implementado usando archivos de texto plano, bases de datos SQL, bases de datos NoSQL, ontología o cualquier forma de almacenamiento o representación. La Figura 4.6 muestra la estructuración de los perfiles de usuario basados en interacción, en forma de diagrama de clases, que se recomienda adoptar para sistemas o aplicaciones que usen la arquitectura de software AALIIoT.

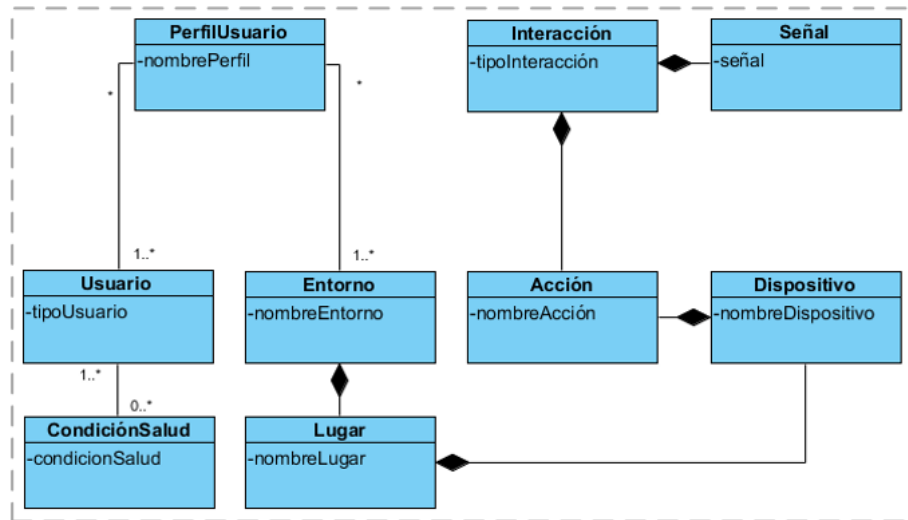


Figura 4.6: Diagrama de clases que define la estructura de un perfil de usuario.  
Fuente: Elaboración propia.

Un usuario puede tener varios perfiles, cada perfil define el tipo de usuarios al que va dirigido, los entornos en los que actúan y las condiciones de salud en las que se encuentran los usuarios que poseen un perfil determinado. Por su parte, los entornos se conforman de lugares (dentro del entorno) que contienen





los dispositivos que los usuarios, pueden controlar, las acciones que pueden ejecutar tales dispositivos y los tipos de interacción habilitados para comunicarse con dichos dispositivos.

Respecto a las interacciones, cada perfil puede definir las señales específicas, habilitadas dentro de un entorno, que representan las solicitudes de ejecución de una acción, por ejemplo, para encender una bombilla de luz, los usuarios con un determinado perfil, pueden emplear una interacción gestual en donde la señal de encendido estará representada por la acción de abrir la mano.

La Tabla 4.1 presenta la descripción de cada uno de los atributos que tiene la estructura de los perfiles de usuario propuesta, los valores que adopten los diferentes atributos deben ser definidos en base a los requerimientos del sistema que implemente la arquitectura de software AALIIoT.

ATRIBUTO	DESCRIPCIÓN	EJEMPLOS
<i>nombrePerfil</i>	Nombre o ID del perfil de usuario	PerfilGeriatrico, PerfilAncianos
<i>tipoUsuario</i>	Tipo de usuario que acepta el perfil.	Anciano, Cuidador
<i>condiciónSalud</i>	Condición de salud de los usuarios.	Discapacidad_física, Diabetes, Ninguna
<i>nombreEntorno</i>	Nombre o ID del entorno para el perfil.	Hogar, Trabajo, Hospital, Escuela
<i>nombreLugar</i>	Nombre o ID del lugar en el entorno para el perfil.	Cocina, Baño, Dormitorio
<i>nombreDispositivo</i>	Nombre o ID del dispositivo objetivo dentro del entorno.	Foco, Radio, TV, Pastillero
<i>nombreAcción</i>	Nombre o ID de la acción que realiza un dispositivo.	ON, OFF
<i>tipoInteracción</i>	Tipo de interacción con la que se solicita ejecutar una acción.	Gestual, GUI, Visual, Vocal, Sensores
<i>señal</i>	Señal que debe realizar el usuario para ejecutar una acción.	Vocal: decir “encender” o “apagar”

Tabla 4.1: Descripción de atributos de la estructura de los perfiles de usuario propuesta.

La Figura 4.7 muestra un ejemplo de la definición de un perfil de usuario implementado como un fichero XML. Este perfil está destinado a todas aquellas personas que son ancianos y presentan una amputación de piernas; el perfil



actúa dentro del hogar, específicamente, en el dormitorio, en donde los usuarios pueden controlar el dispositivo denominado “FOCO1” mediante una interacción gestual, para ello, se han definido las señales específicas que deben realizar los usuarios para encender y apagar el dispositivo.

De esta manera, el definir perfiles de usuarios puede ayudar a resolver un desafío clave dentro de los sistemas informáticos dominantes, el cual consiste en proporcionar la información “correcta” para el usuario “correcto” en el momento “correcto” y en el modo “correcto” [29].

```
<perfil nombrePerfil="Perfil_Dormitorio">
  <usuarios>
    <usuario tipoUsuario="Anciano">
      <condicionesSalud>
        <condicionSalud>Amputación de piernas</condicionSalud>
      </condicionesSalud>
    </usuario>
  </usuarios>
  <entornos>
    <entorno nombreEntorno="Hogar">
      <lugares>
        <lugar nombreLugar="Dormitorio">
          <dispositivos>
            <dispositivo nombreDispositivo="FOCO1">
              <acciones>
                <accion nombreAccion="ON">
                  <interacciones>
                    <interaccion tipo="Gestual">
                      <señal>C:\Gestos\mano_abierta.png</señal>
                    </interaccion>
                  </interacciones>
                </accion>
                <accion nombreAccion="OFF">
                  <interacciones>
                    <interaccion tipo="Gestual">
                      <señal>C:\Gestos\mano_cerrada.png</señal>
                    </interaccion>
                  </interacciones>
                </accion>
              </acciones>
            </dispositivo>
          </dispositivos>
        </lugar>
      </lugares>
    </entorno>
  </entornos>
</perfil>
```

Figura 4.7: Ejemplo de la definición de un perfil de usuario en formato XML.  
Fuente: Elaboración propia.

## 4.3.2. Componentes del Módulo Receptor de Interacción

### 4.3.2.1. Dispositivo receptor de interacción

#### Responsabilidades

Se encarga de recibir los comandos enviados por el usuario. Estos comandos representan las solicitudes para que un dispositivo ejecute cierta acción. Los dispositivos se encuentran monitoreando constantemente el entorno en espera de recibir un comando por parte del usuario. Tales dispositivos pueden ser de diferentes tipos y fabricantes capaces de percibir una o varias formas de interacción, por ejemplo, la Figura 4.8 muestra tres tipos de dispositivos que suelen ser usados para receptar señales gestuales o de movimiento y traducirlo en comandos.



Figura 4.8: Receptores de señales gestuales. a) Kinect b) Brazaletes MYO c) Dispositivo Leap. Fuente: [51], [57], [48].

#### Interacciones

- **Middleware de Interacción:** El Dispositivo receptor de interacción envía la señal captada correspondiente al comando solicitado por el usuario al Middleware de Interacción para que éste, mediante el Controlador de interacción correspondiente al tipo de señal, la procese e interprete. Además, el Identificador de dispositivo objetivo busca dentro de la señal enviada detalles del Dispositivo objetivo a la que va dirigida.
- **Registro de dispositivos receptores de interacción:** Este registro almacena las propiedades de los Dispositivos receptores de interacción para su monitoreo y control.



## Interfaces

Las interfaces de comunicación que proveen los dispositivos receptores de interacción están definidas por sus fabricantes. Es necesario estudiarlas para conocer las funcionalidades que exponen, información que transmiten, formato de datos, métodos de conexión, entre otros. En general, como datos de salida, cada dispositivo envía al Identificador de comandos, dentro del Middleware de Interacción, los comandos correspondientes al tipo de interacción habilitada (comandos de voz, comandos gestuales, comandos mentales, acciones realizadas en GUIs o valores medidos por sensores).

## Técnicas de implementación

Es necesario conocer y entender el dispositivo receptor de interacción para determinar la forma en que debe ser conectado al sistema o aplicación que se vaya a desarrollar, es decir, conocer sus interfaces de comunicación para integrarlo con los demás componentes.

### 4.3.2.2. Registro de dispositivos receptores de interacción

#### Responsabilidades

Puede ser visto como fichero de configuración. En él se detallan las propiedades y especificaciones de cada uno de los dispositivos usados en el entorno para captar la interacción con el usuario.

#### Interacciones

- **Middleware de Interacción:** El middleware accede a este registro para conocer las propiedades del dispositivo que le envía la señal captada en caso de que dichas propiedades no estén expuestas de forma clara.

## Interfaces

Si el componente no actúa como fichero de configuración, entonces será necesario que el componente exponga una interfaz de comunicación a través de la cual se pueda acceder a información como:

- Tipo de señal que envía el dispositivo.
- Formato de datos.
- Tipo de conexiones que admite.
- Puertos de conexión, entre otras.



## Técnicas de implementación

Este componente puede ser implementado como un fichero de configuración (por ejemplo, un fichero XML) en donde cada dispositivo tenga especificadas las propiedades que deben ser conocidas por el Middleware de Interacción. Sin embargo, la definición de requerimientos del sistema que se basará en esta arquitectura puede especificar otra forma de implementación. Además, si los dispositivos receptores de interacción son capaces de identificarse a sí mismos sin la necesidad de estar especificados en un registro, entonces este componente puede ser omitido.

### 4.3.3. Componentes del Middleware de Interacción

El Middleware de Interacción puede ser visto como un componente que está formado por dos sub-componentes auxiliares independientes con funciones específicas. Por esta razón, es necesario describir su funcionalidad en términos de sus responsabilidades, interacciones, interfaces y técnicas de implementación.

#### 4.3.3.1. Middleware de Interacción

##### Responsabilidades

A través de los componentes que lo conforman, procesa e interpreta las órdenes de los usuarios según el tipo de interacción que usan; además, identifica los dispositivos o actuadores sobre los cuales las solicitudes de los usuarios deben tener efecto y genera las solicitudes de ejecución en las cuales se especifica el comando a ejecutar y el dispositivo que llevará a cabo la acción y otros datos que el sistema o aplicación considere relevantes.

##### Interacciones

- **Identificador de usuario:** El Middleware de Interacción se comunica con el Identificador de usuario para determinar las interacciones a las cuales un usuario específico puede acceder, bloqueando aquellas que no le fueron habilitadas.
- **Controlador de ejecución:** El middleware envía al controlador la solicitud de ejecución del comando interpretado y el Dispositivo objetivo al que va dirigido.



## Interfaces

Es necesario que el middleware defina una interfaz de comunicación que permita enviar al Controlador de ejecución datos como:

- Comando interpretado.
- Dispositivo objetivo al que va dirigido.
- Tiempo en el cual se debe llevar a cabo la ejecución.
- Ajuste o auto-ajuste de propiedades para sensores en base a valores medidos.

## Técnicas de implementación

El middleware puede ser implementado como un subsistema que expone servicios Web para la comunicación con el Controlador de ejecución. De esta manera, los envíos de órdenes de ejecución pueden ser optimizadas usando, por ejemplo, cadenas en formato JSON cuyos parámetros representan a las solicitudes del usuario.

A continuación, se describe la funcionalidad de los dos componentes que forman parte del Middleware de Interacción:

### 4.3.3.2. Controlador de interacción

#### Responsabilidades

Se encarga de recibir los diferentes tipos de señales enviadas por los dispositivos usados para receptar los comandos enviados por los usuarios, luego las procesa mediante algoritmos predefinidos, finalmente las interpreta obteniendo así la orden de ejecución que el usuario ha solicitado. Debido a los diferentes tipos de interacción admitidos por la arquitectura, este módulo está formado por varios subcomponentes que definen sus propios algoritmos de procesamiento de comandos según la interacción usada y cuenta con repositorios que almacenan los comandos predefinidos mediante los cuales se puede comparar la señal que está siendo procesada e interpretar la orden del usuario.

## Interacciones

- **Middleware de Interacción:** los datos que genera el Identificador de comandos, luego de procesar e interpretar los comandos recibidos, son tomados por el middleware para formar la solicitud de ejecución.



## Interfaces

Las interfaces de comunicación son provistas por el Middleware de Interacción del cual forma parte este componente.

## Técnicas de implementación

Como ya se había mencionado, este componente está formado por sub-componentes encargados de procesar e interpretar los comandos recibidos según el tipo de interacción: comandos gestuales, mentales, voz, visuales, etc. Cada sub-componente define algoritmos que se ocupan de un tipo de interacción en particular. Los controladores de interacción pueden actuar como servicios internos o externos al Middleware de Interacción.

En el caso de los controladores externos, estos pueden ser anexados al middleware usando cualquier paradigma de comunicación, esto con la finalidad de (re)utilizar algoritmos óptimos para el procesamiento de señales que, en muchos casos, están desarrollados usando herramientas heterogéneas. Los controladores de interacción internos, forman parte de la estructura interna de Middleware de Interacción y se implementan usando las mismas herramientas de desarrollo del middleware. En la Sección 4.3.5, se describen cada uno de los controladores de interacción que la arquitectura admite.

### 4.3.3.3. Identificador de dispositivo objetivo

#### Responsabilidades

Su función principal es la de detectar el Dispositivo objetivo sobre el cual debe tener efecto el comando enviado por el usuario. Además, puede realizar acciones como, por ejemplo, la consulta al Controlador de ejecución sobre el estado de un Dispositivo objetivo, la lista de dispositivos disponibles, los comandos que acepta, etc.

#### Interacciones

- **Middleware de Interacción:** el middleware toma la información generada por este componente a cerca del Dispositivo objetivo para incluirla en la solicitud de ejecución.
- **Controlador de ejecución:** el Identificador de dispositivo objetivo consulta las propiedades del Dispositivo objetivo especificado en la solicitud del middleware para conocer si este acepta los comandos que se desean enviar, su estado actual, etc.



## Interfaces

Las interfaces de comunicación son provistas por el Middleware de Interacción del cual forma parte este componente.

## Técnicas de implementación

Es posible que este componente deba trabajar en paralelo con el Controlador de interacción puesto que los comandos y los dispositivos objetivos pueden venir incluidos en una sola solicitud, por ejemplo, un usuario envía el comando de voz “abrir ventana”, aquí se nota que el comando es “abrir” y el dispositivo objetivo es “ventana”, por ende, los algoritmos que se definen para interpretar los comandos pueden trabajar en conjunto con los algoritmos encargados de identificar los dispositivos o actuadores objetivos.

### 4.3.4. Componentes del Módulo Controlador de Ejecución

#### 4.3.4.1. Controlador de ejecución

##### Responsabilidades

Se encarga de recibir los parámetros de las solicitudes de ejecución y enviar los comandos a los dispositivos objetivo para que estos lo ejecuten.

##### Interacciones

- **Middleware de Interacción:** el middleware le envía las solicitudes de los comandos que interpretó y el Dispositivo objetivo al que va dirigido.
- **Registro de dispositivos objetivo:** el controlador consulta en el registro las propiedades del Dispositivo objetivo especificado en la solicitud del middleware para conocer el formato de datos que admite y su dirección en la red.
- **Dispositivo objetivo:** el controlador le envía el comando al Dispositivo objetivo para que sea ejecutado.

## Interfaces

Las interfaces de entrada reciben datos que provienen del Middleware de Interacción, y corresponde a información como:

- Comando a ejecutar





- Dispositivo objetivo

Por otro lado, el controlador recupera del Registro de dispositivos objetivo datos relacionados a:

- Dirección física del dispositivo en la red
- Comandos admitidos.
- Estado actual
- Disponibilidad, etc.

La interfaz de salida se ocupa de enviar al Dispositivo objetivo el comando recibido desde el middleware para su ejecución.

### Técnicas de implementación

Este componente tiene comunicación directa con los dispositivos o actuadores presentes en el entorno, es por ello que debe ser implementado usando lenguajes de programación que faciliten la comunicación a más bajo nivel evitando el excesivo uso de recursos computacionales. El componente debe ser capaz de enviar los comandos en el formato que el Dispositivo objetivo entiende, por ejemplo, un actuador que enciende una bombilla recibe números enteros: “0” para apagarla y “1” para encenderla.

#### 4.3.4.2. Registro de dispositivos objetivo

##### Responsabilidades

Puede actuar como fichero de configuración. En él se especifican las propiedades y especificaciones de cada uno de los dispositivos o actuadores presentes en el entorno sobre los cuales se van a ejecutar los comandos enviados por los usuarios.

##### Interacciones

- **Controlador de ejecución:** El Registro de dispositivos objetivo brinda a este controlador información correspondiente a los comandos que cada dispositivo es capaz de procesar, la dirección de red a donde debe enviar el comando, el formato de datos que admite, los comandos que acepta, etc.
- **Dispositivo objetivo:** El Registro de dispositivos objetivo requiere que las propiedades y características de cada dispositivo sean registradas.



## Interfaces

Si el componente no actúa como fichero de configuración, es necesario que el componente exponga una interfaz de comunicación a través de la cual se pueda acceder a información como:

- Formato de datos que recibe el Dispositivo objetivo
- Tipo de conexiones que admite
- Dirección en la red y puertos de conexión
- Comandos que acepta, entre otras.

## Técnicas de implementación

El componente puede ser implementado como un fichero de configuración (por ejemplo, un fichero XML) en donde cada Dispositivo objetivo tenga especificadas las propiedades que deben ser conocidas por los componentes con los que interactúa, una base de datos SQL o NoSQL o cualquier forma de almacenamiento.

### 4.3.4.3. Dispositivo objetivo

#### Responsabilidades

Se encarga de llevar a cabo la ejecución del comando solicitado, hace referencia a todos los dispositivos, sensores o actuadores dentro de una red IoT sobre los cuales tendrán efecto los diferentes comandos, por ejemplo, bombillas eléctricas, ventanas eléctricas, puertas eléctricas, alarmas, cucharas inteligentes, pastilleros inteligentes, calefactores, sensores, etc.

#### Interacciones

- **Controlador de ejecución:** El Dispositivo objetivo recibe el comando de ejecución de este componente para hacer efectiva la solicitud.
- **Registro de dispositivos objetivo:** El Dispositivo objetivo debe estar incluido en el registro para que sus propiedades y características puedan ser expuestas.

## Interfaces

Respecto a la comunicación, este componente recibe, como datos de entrada, los comandos que deberá ejecutar.



## **Técnicas de implementación**

Cada Dispositivo objetivo deberá ser implementado considerando las especificaciones del fabricante. O bien, podrían ser modificados mediante programación de sus chips controladores con un lenguaje de programación de bajo nivel para adecuarlos según las necesidades de los sistemas o aplicaciones para entornos AAL. Además, todos los dispositivos controlables dentro del entorno deben formar parte de una red de IoT.

### **4.3.5. Descripción de los controladores de interacción**

En esta sección se describe la funcionalidad de los controladores de interacción que forman parte del Middleware de Interacción. También se da una descripción de los repositorios presentes en el componente mencionado.

#### **4.3.5.1. Controlador de interacción vocal**

Este controlador define el proceso necesario para identificar los comandos de ejecución solicitados por el usuario a través de comandos de voz o sonidos. Los algoritmos usados para identificar lo que una persona dice se fundamentan usando técnicas de *speech recognition* que, a su vez, se basa mayormente, en el uso de redes neuronales [34].

Dentro del entorno, las personas pueden hablar con otras personas constantemente, esto puede significar un problema para los dispositivos que capten comandos vocales pues, no todo lo que una persona expresa con su voz puede ser un comando. Debido a esto, se deben buscar técnicas y métodos para que los usuarios envíen sus comandos de manera adecuada, por ejemplo, el usuario debe comenzar con una entrada como: “comando”, a continuación, debe decir la orden: “encender” y finalmente el dispositivo o dispositivos al que va dirigido, por ejemplo: “luces”. Esto da como resultado la frase “comando encender luces”. De esta manera el controlador de interacción vocal sabe que todo lo que precede a “comando” debe ser procesado como la solicitud del usuario.

#### **4.3.5.2. Controlador de interacción visual**

Este controlador define los algoritmos necesarios para la identificación de comandos usando la vista. Tal reconocimiento puede hacerse mediante seguimiento ocular, es decir, mediante un *eyetracker* se da seguimiento a la mirada del usuario y, cuando realice alguna acción como: posar la mirada en un objeto por un tiempo específico, parpadear, guiñar un ojo, etc., esta sea enviada al controlador de interacción visual para que realice el procesamiento necesario y obtenga el comando al que la acción realizada representa.

#### 4.3.5.3. Controlador de interacción gestual

Este controlador define los algoritmos necesarios para identificar los comandos enviados por el usuario. Dichos comandos pueden estar basados en gestos realizados usando las manos, a través de gestos faciales, o mediante la realización de movimientos corporales específicos. En [65], se presenta un arquitectura de un sistema de control ambiental basada en gestos que se puede tomar como referencia para la implementación de este controlador, además define una serie de gestos que pueden ser usados para solicitar la ejecución de ciertos comandos, mismo que se muestran en la Figura 4.9.

ID	1	2	3	4	5	6	7
FUNCIÓN	Listo	Selección	Salir	Arriba	Abajo	Izquierda	Derecha
GESTO							
ID	8	9	10	11	12	13	14
FUNCIÓN	Borrar	Navegar	Encender	Apagar	Tomar	Soltar	Rotar
GESTO							

Figura 4.9: Gestos relacionados con acciones. Fuente: [65].

#### 4.3.5.4. Controlador de interacción mental

El control del entorno usando señales cerebrales se fundamenta en el uso de interfaces cerebro-computador (BCI, del inglés *Brain Computer-Interface*). BCI se fundamenta en la lectura de las señales cerebrales surgidas durante la sinapsis neuronal generadas en el cerebro de una persona durante la presencia de un pensamiento, procesa tales señales para obtener la intención de la persona y finalmente las convierte en comandos [85]. La Figura 4.10 presenta la arquitectura básica del funcionamiento de aplicaciones BCI, esta arquitectura puede ser tomada como referencia para la implementación de este controlador.

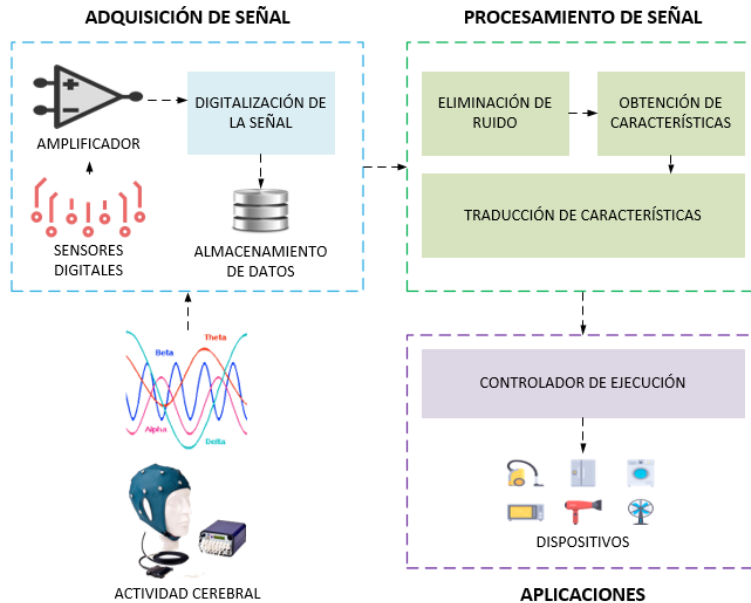


Figura 4.10: Arquitectura que describe el funcionamiento básico de BCI. Fuente: Elaboración propia.

#### 4.3.5.5. Controlador de interacción con sensores

La interacción con sensores consiste en la medición de valores correspondientes a información vital de las personas presentes en el entorno o información del ambiente con la finalidad de ejecutar ciertas acciones cuando se detecten valores previamente especificados. Por ejemplo, cuando la temperatura corporal de un paciente decrece, se pueden emitir alarmas y/o regular la temperatura ambiental. Este controlador requiere procesar los diferentes tipos de datos que recibe de los distintos sensores y conocer cuándo solicitar la ejecución de comandos y sobre qué dispositivos se efectuará la solicitud de ejecución. Para ello, es necesario que este controlador se comunique con algún repositorio en donde se almacene tal información.

#### 4.3.5.6. Controlador de interacción GUI

Este controlador se encarga de gestionar las acciones realizadas en una interfaz gráfica de usuario. Dichas acciones se traducen en comandos que requieren ser ejecutados por parte del Dispositivo objetivo al que van dirigidos.

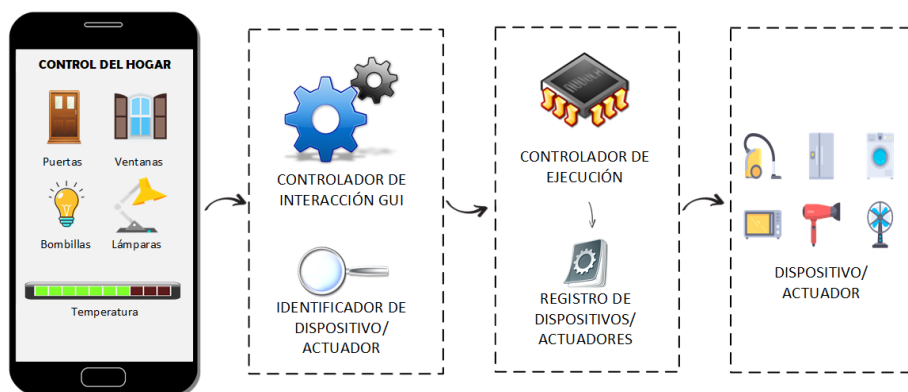


Figura 4.11: Proceso que describe el funcionamiento del control de interacción con GUIs. Fuente: Elaboración propia.

El proceso presentado en la Figura 4.11 puede ser tomado como referencia para implementar este controlador. La ventaja de tener un controlador de interacción GUI es evidente cuando se desarrollan diferentes tipos de interfaces gráficas para controlar los dispositivos dentro de un entorno en particular. Por ejemplo, para el control de los dispositivos presentes en un hogar, se pueden desarrollar interfaces gráficas que se despliegan en ordenadores de escritorio, o aplicaciones para *smartphones* con diferentes sistemas operativos, todas las acciones realizadas serán enviadas a este controlador que se encargará de convertirlas en comandos para su posterior ejecución.

#### 4.3.5.7. Repositorios de comandos

Estos repositorios almacenan información adicional útil para que los diferentes controladores de interacción puedan comparar un comando recibido con la información de los comandos guardada en los repositorios y determinar a cuál corresponde. Por ejemplo, el Repositorio de comandos gestuales almacena las imágenes de todos los gestos admitidos e información de las acciones que ejecuta, el Controlador de interacción gestual procesa la imagen del gesto realizado por el usuario y busca su existencia en el repositorio, cuando la encuentre, recupera la información de la acción que corresponde a tal gesto, este proceso se muestra en la Figura 4.12. De la misma manera interactúan los demás controladores con sus respectivos repositorios.

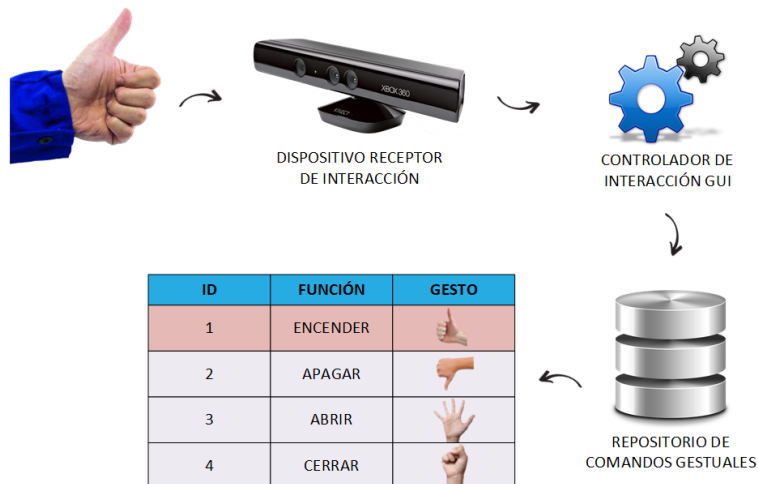


Figura 4.12: Búsqueda del gesto realizado por el usuario en el repositorio de comandos gestuales. Fuente: Elaboración propia.

#### 4.3.5.8. Repositorio de información de salud

Este repositorio almacena toda la información de salud recopilada mediante la interacción con sensores, puede ser visto como varios sub-repositorios interrelacionados. Por ejemplo, un repositorio almacena los valores recogidos por sensores que monitorean los signos vitales de una persona, otro repositorio almacena información ambiental. Estos repositorios son útiles tanto para cuidadores como para doctores ya que brindan información indispensable para realizar un monitoreo, remoto o personal, de la salud de las personas. Además, es necesaria la existencia de un sub-repositorio que almacene información de las acciones que deben ser ejecutadas cuando los sensores obtengan ciertos valores previamente especificados, esto para activar alarmas, regular el funcionamiento de dispositivos como: calefactores, intensidad de la luz, etc.

Este repositorio de información de la salud de las personas debe poseer una correlación con el Repositorio de información del usuario, de manera que el registro de cada usuario se corresponda con registros en el repositorio de salud.

#### 4.3.6. Protocolos de comunicación

Los protocolos de comunicación van a permitir definir un formato y un conjunto de reglas comunes para intercambiar mensajes entre los dispositivos



que formen parte de los sistemas o aplicaciones basados en la arquitectura de software propuesta. Para habilitar la comunicación entre dos dispositivos, no basta con realizar una conexión física, además, es necesario definir la forma en la que se llevará a cabo la comunicación.

La arquitectura de software propuesta, no se limita al uso de un protocolo de comunicación y transmisión de datos en particular. El protocolo a usar, dependerá de los requerimientos funcionales del sistema, mismos que deberán ser definidos por el equipo de analistas y desarrolladores de software. A manera de recomendación, se considera importante usar protocolos bien conocidos y probados por la comunidad ya que esto facilita el acceso a una gran cantidad de documentación y acelera los tiempos de desarrollo. En el Capítulo 3, como resultados del mapeo sistemático, se obtuvo que los protocolos de comunicación más usados en aplicaciones IoT dentro del área de la salud para la conexión y transmisión de datos entre dispositivos, sensores y actuadores son: *WiFi*, *Bluetooth* y *Zigbee*.

Se pueden combinar varios protocolos de comunicación dentro del mismo sistema o aplicación, por ejemplo, la comunicación entre los diferentes módulos de la arquitectura se puede llevar a cabo usando protocolos de red inalámbricos como *WiFi* y la comunicación entre los dispositivos, sensores y actuadores se puede desarrollar mediante protocolos de comunicación propios de IoT como *Zigbee*. Nuevamente, esta decisión deberá ser tomada en base a los requerimientos del sistema a implementar.

#### 4.3.7. Diagrama de componentes

En esta sección se presenta el diagrama de componentes de la arquitectura de software propuesta (Figura 4.13). El diagrama permite visualizar la interacción que existe entre cada uno de los componentes mediante las interfaces que definen y las diferentes dependencias entre componentes y subcomponentes.

En el diagrama de componentes, el Middleware de Interacción es mostrado como el componente principal ya que se comunica con la mayoría del resto de componentes de la arquitectura. El middleware intercambia datos con los siguientes componentes:

- Identificador de usuarios.
- Dispositivo receptor de interacción.
- Registro de dispositivos receptores de interacción.
- Registro de dispositivos objetivo.



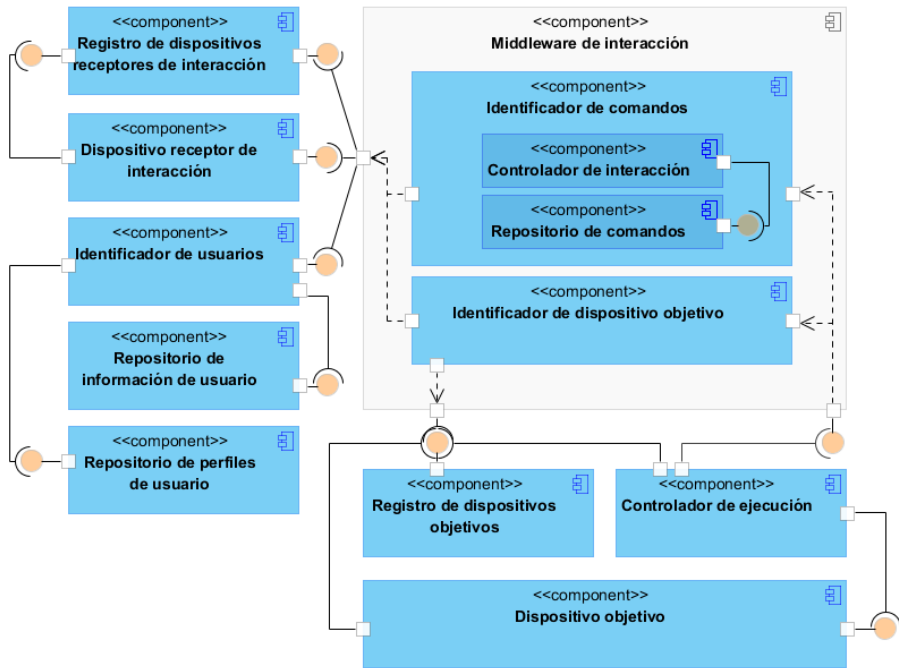


Figura 4.13: Diagrama de componentes de la arquitectura AALIoT. Fuente: Elaboración propia.

Además, el middleware está compuesto por sub-componentes que permiten llevar a cabo la identificación de comandos y la identificación de dispositivos objetivo; los resultados que obtiene de estos subcomponentes los expone en una interfaz a la cual accede el Controlador de ejecución para recuperar los comandos identificados.

## 4.4. Vista dinámica

Esta sección describe los aspectos dinámicos de la arquitectura de software propuesta. Para realizar dicha descripción, se emplean diagramas de actividad UML, ya que estos ayudan a mostrar, a un alto nivel, la forma en la que se construyen e inician los diferentes flujos de trabajo y/o procesos y los puntos en dónde puede ocurrir el procesamiento paralelo.



#### 4.4.1. Diagrama de actividad del Módulo Gestor de Usuarios

El diagrama de actividad mostrado en la Figura 4.14, presenta el flujo de trabajo que deben llevar a cabo los componentes de este módulo para identificar a un usuario y recuperar su perfil. En este diagrama está presente el componente Dispositivo receptor de interacción pues, cuando se lleva a cabo la autenticación, el usuario interactúa con algún dispositivo que recoge sus datos para que estos puedan ser procesados por el Identificador de usuarios.

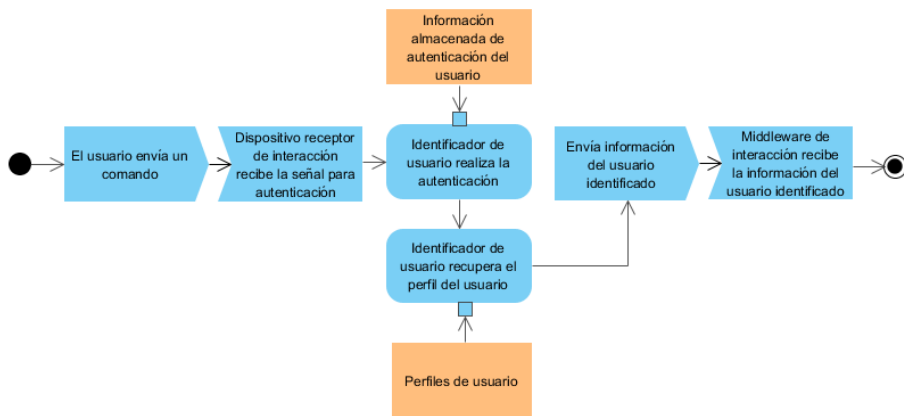


Figura 4.14: Diagrama de actividad del Módulo Gestor de Usuarios. Fuente: Elaboración propia.

#### 4.4.2. Diagrama de actividad del Módulo Receptor de Interacción

El diagrama de actividad mostrado en la Figura 4.15, presenta el flujo de tareas que debe llevar a cabo el Dispositivo receptor de interacción para recibir los comandos solicitados por el usuario en forma de señales de diversos tipos (voz, gestos, pensamientos, visuales, etc.) y enviarlos al Middleware de Interacción para su respectiva interpretación.

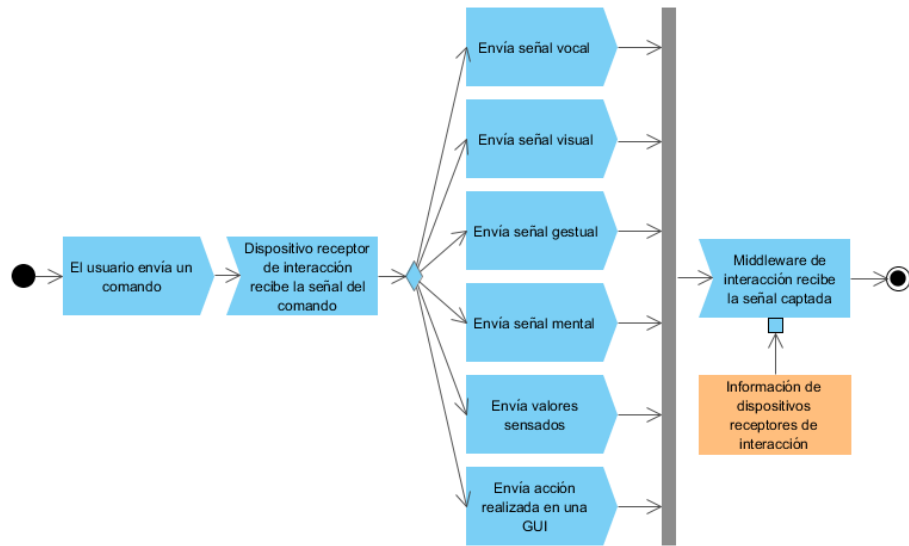


Figura 4.15: Diagrama de actividad del Módulo Receptor de Interacción. Fuente: Elaboración propia.

La identificación de usuarios, dependiendo de los requerimientos del sistema que empleará la arquitectura propuesta, puede ser un proceso que se realice en paralelo con la interpretación de los comandos enviados por un usuario, es decir, para interpretar un comando es necesario realizar, al mismo tiempo, la identificación del usuario que lo envió y su perfil de usuario.

En la Figura 4.16 se presenta el diagrama de actividad que modela la situación mencionada, desde el envío de comandos por parte del usuario hasta la recepción del Middleware de Interacción de las diversas señales generadas.

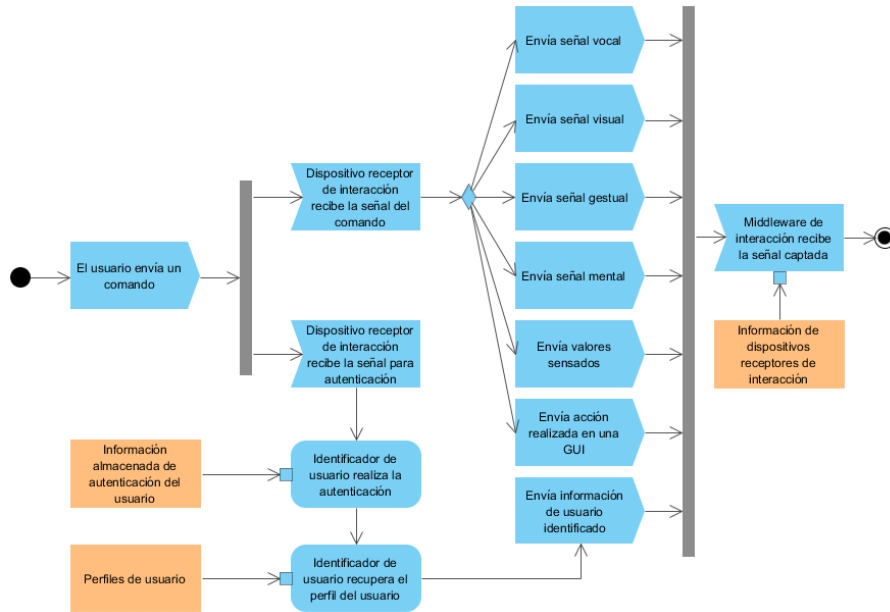


Figura 4.16: Diagrama de actividad para captar interacción e identificar usuarios. Fuente: Elaboración propia.

#### 4.4.3. Diagrama de actividad del Middleware de Interacción

La Figura 4.17 presenta el diagrama de actividad que muestra el flujo de tareas que lleva a cabo el Middleware de Interacción para recibir la señal correspondiente a un comando, procesar e identificar dicha señal para obtener el comando solicitado, identificar el Dispositivo objetivo y generar la solicitud de ejecución. Se observa que el Controlador de interacción y el Identificador de dispositivo objetivo son componentes que se pueden ejecutar de forma paralela, estos componentes generan información necesaria para formar la solicitud de ejecución para que el Controlador de ejecución la procese adecuadamente.

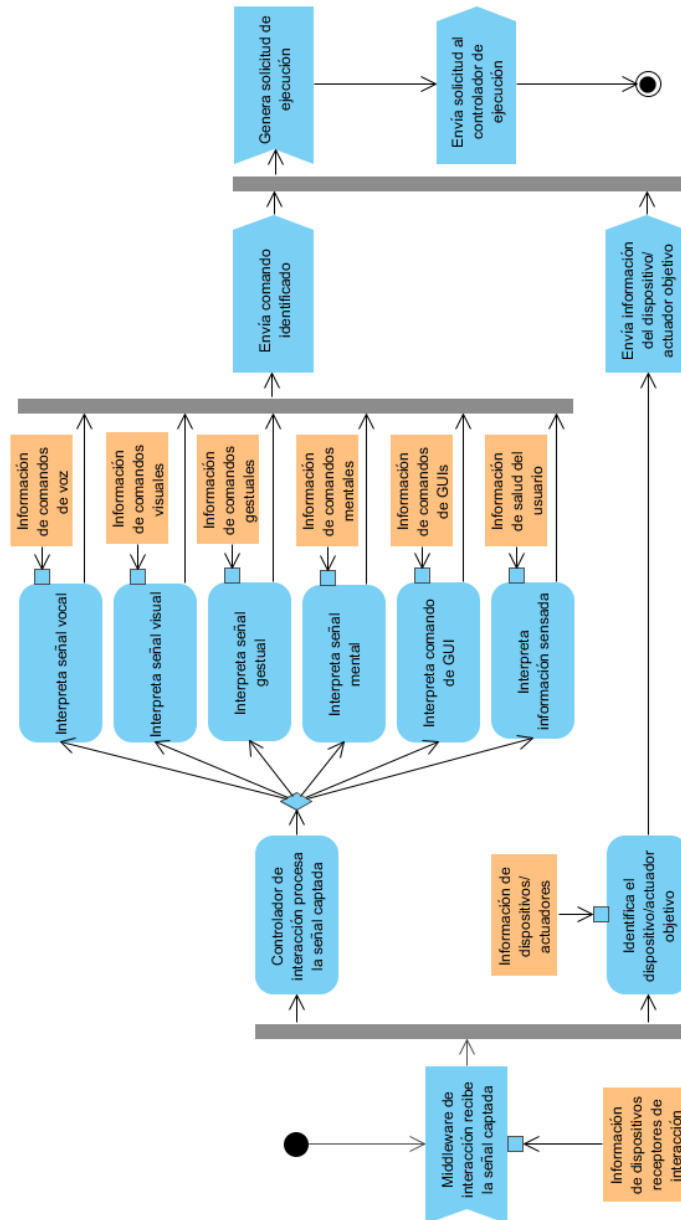


Figura 4.17: Diagrama de actividad del Middleware de Interacción. Fuente: Elaboración propia.



#### 4.4.4. Diagrama de actividad del Módulo Controlador de Ejecución

Una vez que el flujo de tareas correspondientes al Middleware de Interacción finaliza, da inicio el flujo de tareas mostrado en el diagrama de actividad del Módulo Controlador de Ejecución mostrado en la Figura 4.18. Las acciones inician cuando el Controlador de ejecución recibe la solicitud y avanzan hasta el punto en el que el Dispositivo objetivo ejecuta la acción solicitada por el usuario pasando por una serie de procesos intermedios.

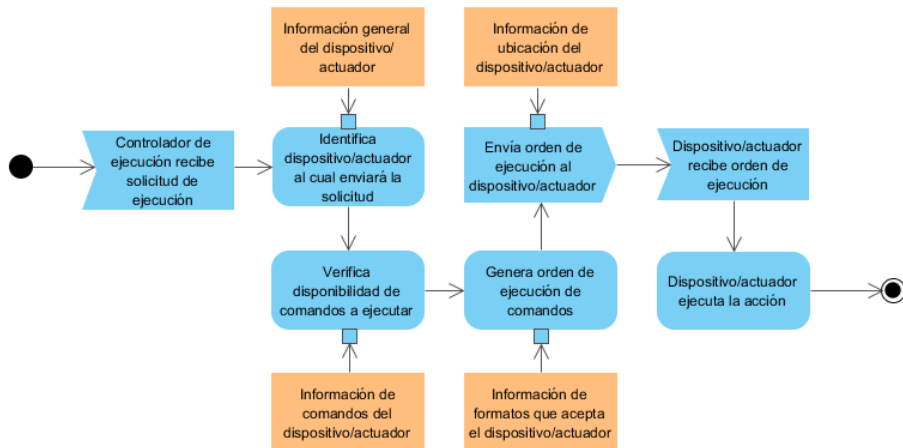


Figura 4.18: Diagrama de actividad del Módulo Controlador de Ejecución.  
Fuente: Elaboración propia.





## Capítulo 5

# Implementación

En este capítulo se presenta la implementación del prototipo de un sistema enfocado en la Interacción Hombre Computador (HCI) para el internet de las cosas (IoT) que se despliega en entornos de vida asistidos (AAL). El prototipo se desarrolla considerando los módulos y componentes que forman parte de la arquitectura de software AALIIoT presentada en el Capítulo 4. El objetivo es presentar al sistema como una instancia de la arquitectura propuesta e ilustrar cómo ésta permite el desarrollo de aplicaciones de software fundamentado en cada uno de sus módulos y componentes.

El prototipo del sistema presentado, es la representación de la solución dada a un caso de estudio basado en un escenario específico. A partir de ese escenario, se obtendrá una perspectiva del sistema de donde se deducirán cada una de sus funcionalidades. En este sentido, la sección 5.1 presenta dicho escenario, la sección 5.2 presenta la perspectiva que se tiene del sistema, la sección 5.3 presenta la descripción del sistema en términos de los requerimientos funcionales, casos de uso y diagramas de secuencia, la sección 5.4 presenta la descripción del sistema detallando cómo fueron desarrollados sus diferentes módulos y componentes, la sección 5.5 presenta el despliegue del sistema y, finalmente, la sección 5.6 presenta las conclusiones de este capítulo.





## 5.1. Escenario

El escenario planteado, representa un adulto mayor quien posee una discapacidad física, él no tiene movilidad en sus dos piernas por lo cual debe usar una silla de ruedas para trasladarse de un lugar a otro. Dicha persona, pasa las noches en su habitación, prefiriendo estar sin la compañía de su cuidador. Muchas veces se levanta a media noche y lee su libro favorito; sin embargo, se le dificulta mucho tener que encender la bombilla de luz de su habitación de forma manual. A esta persona, le parecería mucho más cómodo y le daría más independencia el poder realizar esta acción sin tener que levantarse de su cama o llamar a su cuidador.

## 5.2. Perspectiva del sistema

El sistema a ser construido, debe facilitar al usuario la estadía en su entorno (dominio de AAL) permitiendo al usuario usar algún tipo de interacción para poder encender una bombilla de luz (dominio de HCI). Además, es necesario incluir un dispositivo receptor de interacción para que capture las señales enviadas por el usuario; luego, se debe re-adaptar las conexiones eléctricas de la bombilla de luz para que ésta pueda ser controlada por algún dispositivo que reciba las solicitudes procesadas del usuario convertidas en comandos y que los ejecute mediante una red inalámbrica, es decir, el dispositivo debe poder conectarse a internet (dominio IoT).

La arquitectura de software AALIIoT presentada en el Capítulo 4, brinda todos los módulos y componentes que pueden ser usados para el diseño e implementación del prototipo del sistema solicitado.

## 5.3. Descripción de la funcionalidad del sistema

Considerando las necesidades de interacción del usuario correspondiente al escenario plantado, se ha determinado que éste podrá tener el control de la bombilla de luz mediante una interacción gestual e interacción con una interfaz gráfica de usuario (GUI). La descripción de la funcionalidad del sistema se presenta mediante la definición de los requerimientos funcionales del usuario, diagramas de casos de uso y diagramas de secuencia.

### 5.3.1. Requerimientos funcionales de usuario (RF)

Los requerimientos han sido desglosados de manera que respondan a cada una de las acciones que estarán disponibles para los usuarios:



- RF1: Encender la bombilla de luz usando un gesto con la mano.
- RF2: Apagar la bombilla de luz usando un gesto con la mano.
- RF3: Encender la bombilla de luz usando una GUI.
- RF4: Apagar la bombilla de luz usando una GUI.

### 5.3.2. Casos de uso

En esta sección se especifican los requerimientos del sistema mediante diagramas de casos de uso. Los casos de uso fueron divididos según el tipo interacción usada. Los actores fueron determinados en base a los módulos principales de la arquitectura propuesta en este trabajo.

La Figura 5.1 muestra todos los casos de uso que son necesarios para cumplir con los requerimientos funcionales de usuario RF1 y RF2.

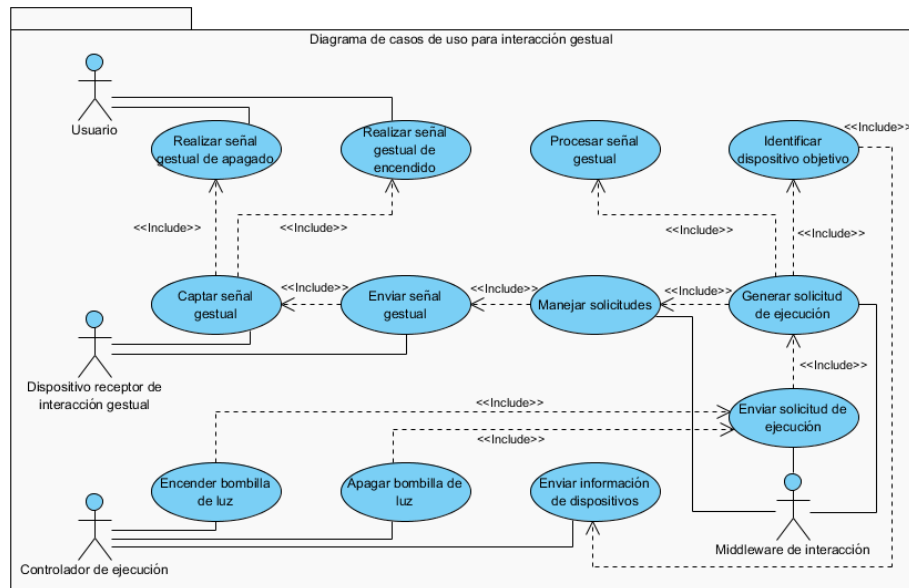


Figura 5.1: Diagrama de casos de uso para interacción gestual. Fuente: Elaboración propia.

La Figura 5.2 muestra todos los casos de uso que son necesarios para cumplir con los requerimientos funcionales de usuario RF3 y RF4.

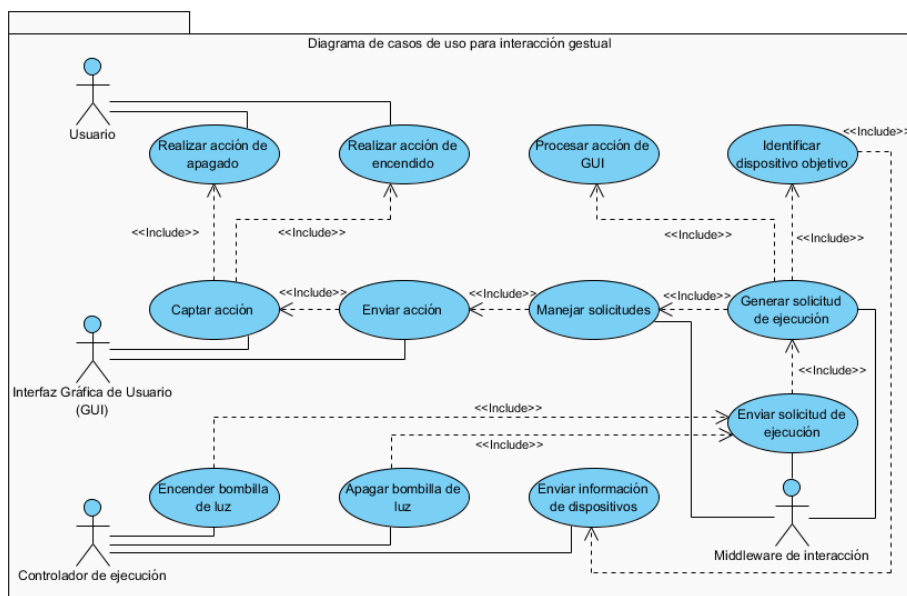


Figura 5.2: Diagrama de casos de uso para interacción con GUIs. Fuente: Elaboración propia.

### 5.3.3. Diagramas de secuencia

En esta sección se presentan los diagramas de secuencia que muestran la interacción de los componentes de los diferentes módulos de la arquitectura del sistema a través del tiempo. Para ello, es necesario definir la perspectiva de implementación que tendrá el sistema y así establecer la manera en la que los mensajes serán transmitidos entre cada componente.

El sistema en cuestión se implementó usando el Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés *Hypertext Transfer Protocol*) usando el paradigma Cliente-Servidor; los recursos serán enviados y recuperados entre los diferentes módulos mediante solicitudes *GET* y *POST* desde los clientes hasta un servidor.

En este sistema, los Controladores de Interacción actúan como clientes que consumen los servicios expuestos por el Middleware de interacción, el cual actúa como servidor, para el envío de las señales convertidas en comandos y los dispositivos sobre los cuales actuarán dichos comandos. Por otro lado, el Middleware de interacción también actúa como cliente ya que consume los servicios expuestos por el Controlador de ejecución para el envío de solicitudes

de ejecución y consulta de información de dispositivos.

## Dispositivo receptor de interacción

La Figura 5.3 muestra el diagrama de secuencia producido cuando el Dispositivo receptor de interacción, según la interacción empleada, captura la señal realizada por el usuario y la envía al Controlador de interacción para su procesamiento e interpretación.

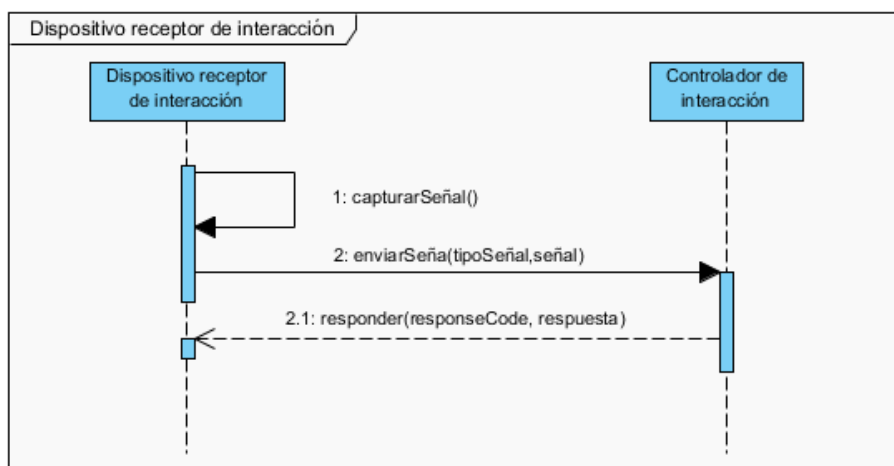


Figura 5.3: Diagrama de secuencia: Dispositivo receptor de interacción. Fuente: Elaboración propia.

## Controlador de interacción

La Figura 5.4 presenta el diagrama de secuencia producido cuando el Controlador de interacción, según la interacción empleada, recibe una señal, la procesa, la interpreta y la envía al Middleware de interacción para que genere la solicitud de ejecución respectiva.

Este Controlador de interacción usa algoritmos que permiten procesar la señal y obtener los parámetros (getuales, vocales, mentales, visuales, etc.) que pueden ser consultados en el repositorio de comandos con la finalidad de obtener el comando al que representan dichos parámetros y el dispositivo al cual van dirigidos. Luego envía la información obtenida al Middleware de Interacción mediante una solicitud HTTP. Finalmente, el Middleware contesta la solicitud mediante un mensaje que incluye un código HTTP y la cadena de respuesta.

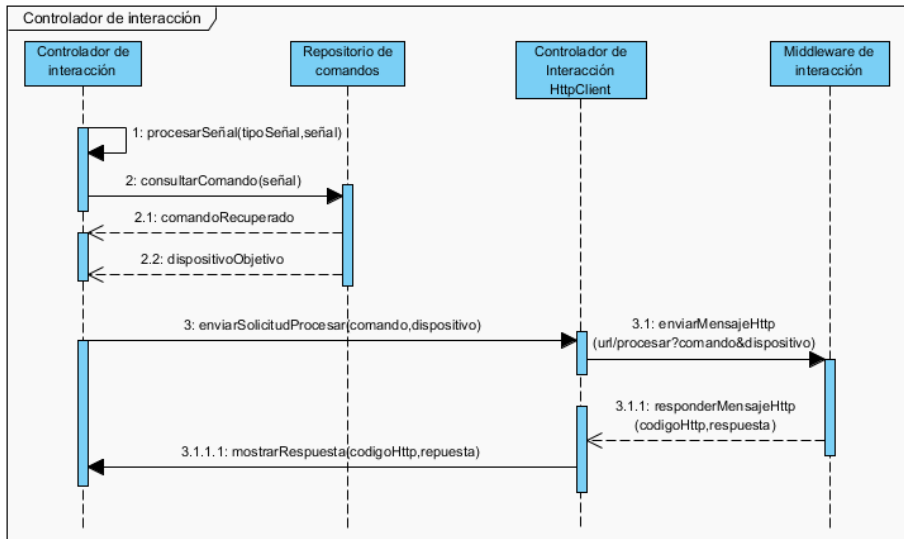


Figura 5.4: Diagrama de secuencia: Controlador de interacción. Fuente: Elaboración propia.

## Identificador de dispositivo objetivo

La Figura 5.5 muestra el diagrama de secuencia producido cuando el Identificador de dispositivo objetivo consulta información al Controlador de ejecución sobre un dispositivo objetivo específico en base a un criterio (comandos que acepta, disponibilidad, estado actual, etc.).

El Identificador de dispositivo envía una solicitud HTTP al Controlador de ejecución especificando los parámetros de consulta, el Controlador de ejecución busca la información solicitada en el Registro de dispositivos y contesta la solicitud con un mensaje que incluye el código HTTP y la información recuperada.

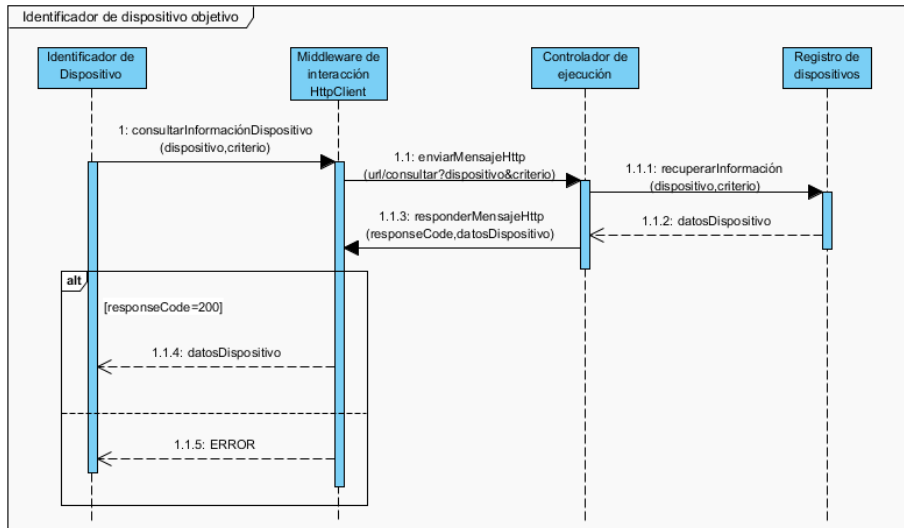


Figura 5.5: Diagrama de secuencia: Identificador de dispositivo objetivo. Fuente: Elaboración propia.

## Middleware de interacción

La Figura 5.6 muestra el diagrama de secuencia que corresponde al envío de una solicitud de ejecución, que consta del comando a ejecutar y el dispositivo objetivo, desde el Middleware de interacción al Controlador de ejecución.

El Middleware de Interacción recibe el mensaje HTTP enviado por el Controlador de interacción luego de que éste haya porcesado la señal realizada por el usuario. Luego de manejar el mensaje, el Middleware genera la solicitud de ejecución y la envía mediante una solicitud HTTP al Controlador de ejecución el cual contestará a dicha solicitud con un código HTTP y la cadena de respuesta.

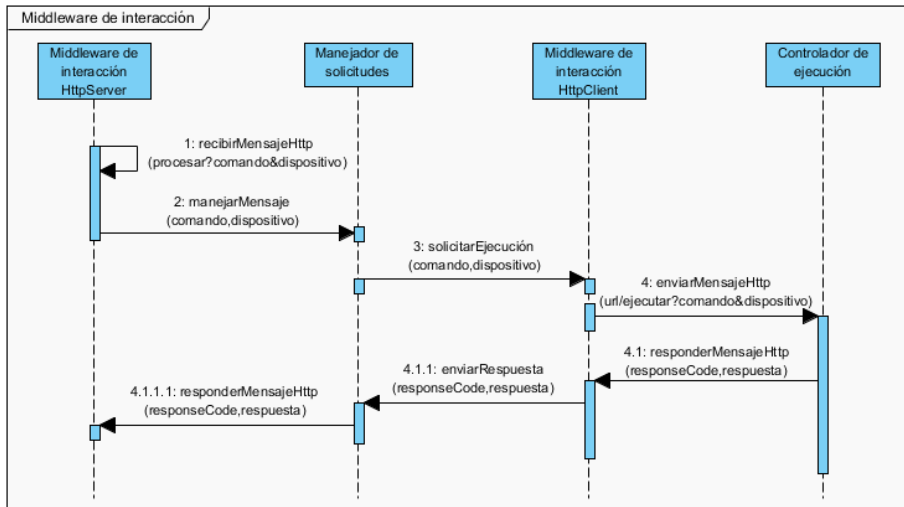


Figura 5.6: Diagrama de secuencia: Middleware de interacción. Fuente: Elaboración propia.

## Controlador de ejecución

La Figura 5.7 muestra el diagrama de secuencia que corresponde a la recepción de una solicitud de ejecución por parte del Controlador de ejecución para encender o apagar una bombilla de luz.

Este controlador recibe la solicitud HTTP que incluye el comando a ejecutar y el dispositivo objetivo, procesa la solicitud y, en base al comando recibido, ejecuta la acción sobre el dispositivo especificado en la solicitud de ejecución. Finalmente, el Controlador de ejecución contestará al componente que envió la solicitud de ejecución mediante un código HTTP y la cadena de respuesta.

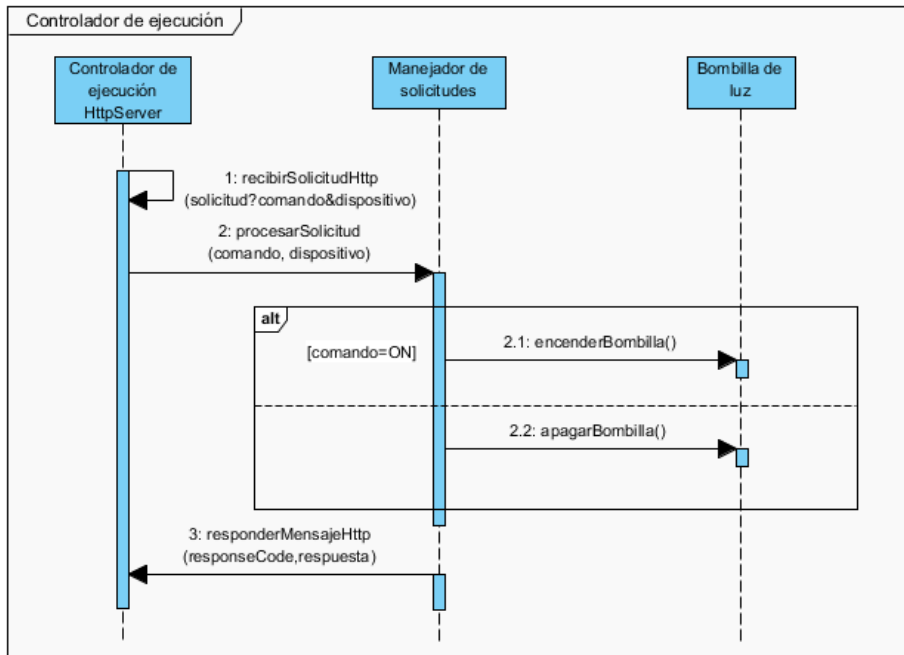


Figura 5.7: Diagrama de secuencia: Controlador de ejecución. Fuente: Elaboración propia.

## 5.4. Implementación del prototipo del sistema

Los módulos del sistema se desarrollaron usando diferentes lenguajes de programación, tales como Python, Java y Arduino. La comunicación entre cada módulo, como se mencionó anteriormente, se llevó a cabo usando el protocolo HTTP a través de sus métodos *GET* y *POST*, es decir, existe un cliente que se comunica con un servidor mediante la invocación de un Localizador Uniforme de Recursos (URL, por sus siglas en inglés, *Uniform Resource Locator*), el servidor procesa la solicitud y devuelve un código HTTP de respuesta (*response code*) y su contenido, este proceso puede ser observado en los diagramas de secuencia presentados en la sección anterior. Todas las herramientas usadas en el proceso de implementación se describen en el Apéndice D.

En la Tabla 5.2 se describe la estructura de las URLs que define el Middleware de Interacción.





SOLICITUD		DETALLE
<i>/test</i>	Descripción	Permite al cliente realizar una prueba de conexión con el Middleware.
	Parámetros	No posee parámetros.
	Ejemplo	<i>http://ip:puerto/test</i>
<i>/procesar</i>	Descripción	El cliente solicita al Middleware el procesamiento de la señal captada.
	Parámetros	- <i>nombreDispositivo</i> : Nombre del dispositivo que envía la solicitud. - <i>tipoInteraccion</i> : Tipo de interacción que maneja el dispositivo que envía la solicitud. - <i>senal</i> : señal captada por el dispositivo que envía la solicitud.
	Ejemplo	<i>http://ip:puerto/procesar?nombreDispositivo=ANDROID_GUI&amp;tipoInteraccion=GUI&amp;senal=ON-FOCO1</i>

Tabla 5.1: Descripción de URLs definidas por el Middleware de Interacción.

En la Tabla 5.2 se describe la estructura de las URLs que define el Controlador de ejecución.

SOLICITUD		DETALLE
<i>/test</i>	Descripción	Permite al cliente realizar una prueba de conexión con el Controlador de ejecución.
	Parámetros	No posee parámetros.
	Ejemplo	<i>http://ip:puerto/test</i>
<i>/notificar</i>	Descripción	El Middleware envía al Controlador de ejecución el comando que el usuario desea ejecutar.
	Parámetros	- <i>dispositivo</i> : Dispositivo objetivo al que va dirigido el comando. - <i>comando</i> : Comando que se desea ejecutar sobre el dispositivo objetivo.
	Ejemplo	<i>http://ip:puerto/notificar?dispositivo=FOCO1&amp;comando=ON</i>
<i>/comandos</i>	Descripción	El Middleware solicita al Controlador de ejecución la lista de comandos aceptados por un dispositivo específico.
	Parámetros	<i>dispositivo</i> : Dispositivo del cual se desea consultar los comandos.



	Ejemplo	<i>http://ip:puerto/comandos?dispositivo=FOCO1</i>
<i>/dispositivos</i>	Descripción	El Middleware solicita al Controlador de ejecución la lista de dispositivos presentes en el entorno.
	Parámetros	No posee parámetros
	Ejemplo	<i>http://ip:puerto/dispositivos</i>
<i>/habilitado</i>	Descripción	El Middleware consulta al Controlador de ejecución si un dispositivo se encuentra habilitado en el entorno.
	Parámetros	<i>dispositivo</i> : Dispositivo del cual se desea consultar los comandos.
	Ejemplo	<i>http://ip:puerto/habilitado?dispositivo=FOCO1</i>

Tabla 5.2: Descripción de URLs definidas por el Controlador de ejecución.

A continuación, se da a conocer el detalle de implementación de cada uno de los módulos del sistema.

### 5.4.1. Módulo receptor de interacción

Este módulo se compone de los dispositivos que captan las señales realizadas por los usuarios para su posterior procesamiento e interpretación.

#### 5.4.1.1. Receptor de Interacción Gestual

Para la interacción gestual, se empleó una cámara de video para capturar las señales que el usuario realiza con cualquiera de sus dos manos, la señal capturada se envía al Controlador de Interacción Gestual para su procesamiento.

#### 5.4.1.2. Receptor de Interacción GUI

Para la interacción GUI, se ha creado una interfaz gráfica de usuario para una aplicación móvil que se despliega en smartphones con sistema operativo Android (Figura 5.8). La aplicación fue implementada usando el paradigma de programación por bloques que brinda el entorno de desarrollo *App Inventor*<sup>1</sup>.

<sup>1</sup><http://ai2.appinventor.mit.edu/>



Figura 5.8: Aplicación móvil que funciona como Receptor de Interacción GUI. a) GUI principal, b) GUI de configuración, c) GUI de control. Fuente: Elaboración propia.

La aplicación consta de tres pantallas, la Figura 5.8a presenta la pantalla principal de la aplicación, la Figura 5.8b presenta la pantalla de configuración, en donde se pueden configurar los parámetros de conexión con el Middleware de Interacción y la Figura 5.8c presenta la pantalla de control, esta pantalla tiene los botones mediante los cuales se controlará el encendido o apagado de la bombilla de luz.

En la Figura 5.9 se presenta la codificación por bloques que permite el establecimiento de los parámetros de conexión con el Middleware de Interacción, dichos parámetros corresponden a la IP y el puerto de conexión en el que está escuchando el Middleware y son ingresados por el usuario en los campos disponibles en la pantalla de configuración. Al pulsar el botón “Conectar”, se envía una solicitud */test* para probar si el Middleware está disponible.

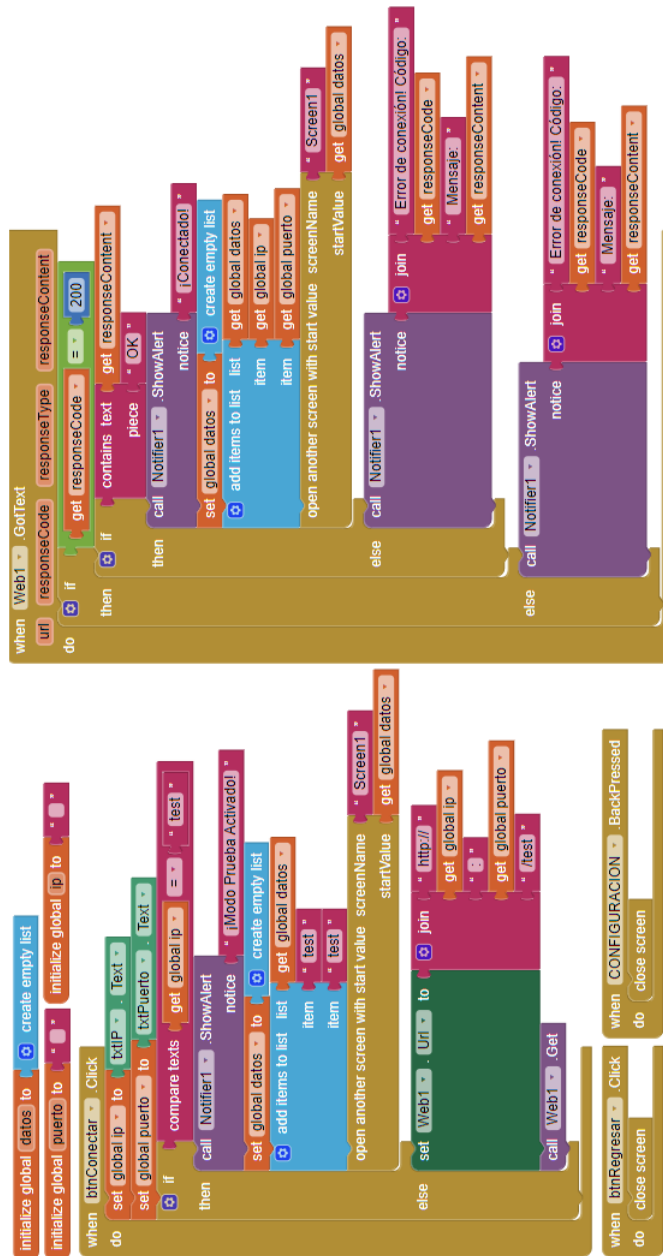


Figura 5.9: Programación por bloques de la GUI de Configuración. Fuente: Elaboración propia.

La programación por bloques de la pantalla de control contiene el código necesario para realizar las solicitudes HTTP a través de las URLs que expone el Middleware. Al presionar cualquiera de los botones (“Encender” o “Apagar”), se invoca la solicitud */procesar* enviando la señal captada al Middleware para que la procese. La Figura 5.10 muestra la programación por bloques que se ejecuta cuando el usuario presiona el botón “Encender”.

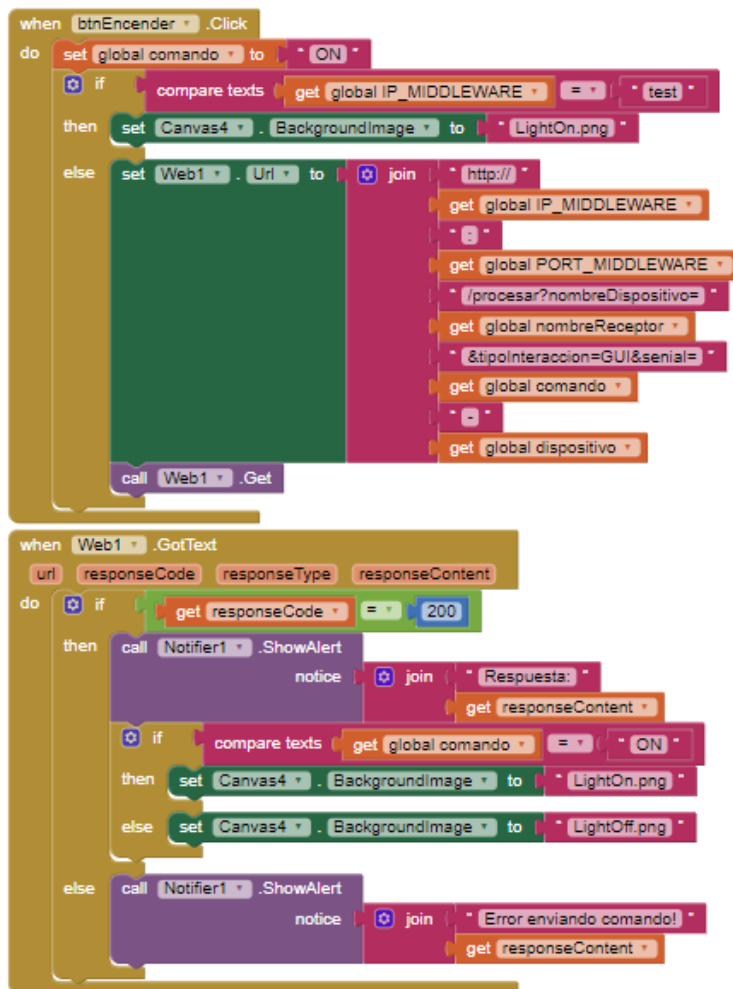


Figura 5.10: Programación por bloques del botón “Encender” de la GUI de control. Fuente: Elaboración propia.



### 5.4.2. Middleware de interacción

Este módulo fue implementado usando el lenguaje de programación Java. La estructura del proyecto se presenta en la Figura 5.11.

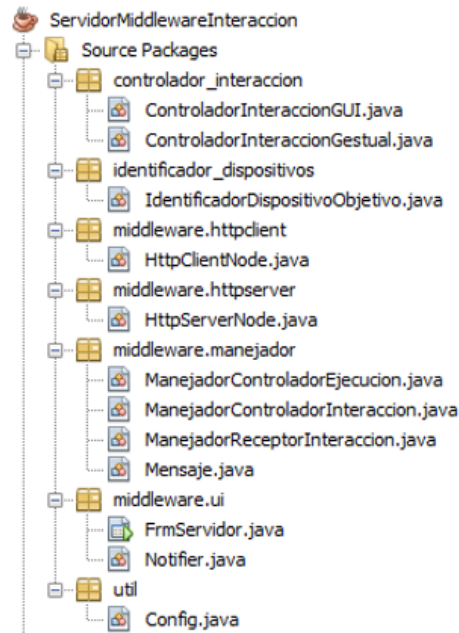


Figura 5.11: Estructura del Middleware de interacción. Fuente: Elaboración propia.

Se puede observar que el paquete “controlador\_interacción” contiene las clases que definen los controladores para cada una de las interacciones habilitadas en este sistema. Por otro lado, la clase “IdentificadorDispositivoObjetivo.java” define los métodos necesarios para consultar información de los dispositivos que el usuario desea controlar: estado actual, comandos que acepta, disponibilidad, etc. La clase “HttpClientNode.java” establece los métodos necesarios para invocar a las URLs definidas en el Controlador de ejecución. Además, la clase “HttpServerNode.java” define los manejadores de las URLs que el Middleware de Interacción expone a sus clientes, mismas que se describen en la Tabla 5.1.

También se puede apreciar que el paquete “middleware.manejador” contiene todos los manejadores que se encarga hacer que las solicitudes enviadas por módulos externos al Middleware sean redireccionadas hacia los componentes adecuados, los mensajes enviados al Controlador de ejecución y sus respuestas.



El middleware posee una interfaz gráfica de usuario en la que se muestran todas las solicitudes entrantes desde módulos externos y las solicitudes que se envían al Controlador de ejecución; además, permite especificar la dirección y el puerto del Controlador de ejecución presente en el entorno a través del menú “Configuración”, dicha interfaz gráfica se muestra en la Figura 5.12.



Figura 5.12: Interfaz gráfica de usuario del Middleware de interacción. Fuente: Elaboración propia.

La especificación de la arquitectura, indica que los controladores de interacción pueden actuar como servicios internos o externos al Middleware de Interacción. En el caso de los controladores externos, éstos pueden ser anexados al Middleware usando cualquier paradigma de comunicación, esto con la finalidad de (re)utilizar algoritmos más óptimos para el procesamiento de señales que, en muchos casos, están desarrollados usando herramientas heterogéneas o son consumidos a través de APIs de servicios *Cloud*.

A continuación, se describen los componentes controladores de interacción, tanto internos como externos, encargados de procesar las señales captadas por los Dispositivos receptores de interacción habilitados en el entorno.

#### 5.4.2.1. Controlador de interacción GUI

Este componente fue implementado como servicio interno del Middleware de Interacción. Recibe las acciones generadas en la interfaz gráfica presentada al usuario para el control de los dispositivos IoT y las transforma en los comandos respectivos.



En la Figura 5.12, se observan los mensajes generados durante la comunicación del Dispositivo receptor de interacción GUI con el Middleware de Interacción y el Controlador de Ejecución; en base al orden en el que se muestran, se puede apreciar lo siguiente:

- Línea 1: Indica que el Middleware se ha iniciado correctamente en el puerto 13000.
- Línea 2: El Middleware recibe una solicitud de prueba de conexión desde un cliente.
- Línea 3: El usuario ha presionado el botón “Encender” en el Dispositivo receptor de interacción GUI (ANDROID\_GUI), este envía la señal al Middleware, en donde el Controlador de interacción GUI toma la señal, la procesa, la interpreta y genera la solicitud de ejecución para encender la bombilla de luz usando el código mostrado en la Figura 5.13.
- Línea 4: El Middleware envía la solicitud de ejecución al Controlador de ejecución (WIFI\_CONTROLLER) indicando el dispositivo objetivo (FOCO1) y el comando que se desea ejecutar (*ON*) mediante el método mostrado en la Figura 5.14.
- Línea 5: Un proceso similar al descrito para la línea 3 ocurre cuando el usuario presiona el botón “Apagar” en el Dispositivo receptor de interacción GUI.
- Línea 6: El Middleware envía la solicitud de ejecución al Controlador de ejecución indicando el dispositivo objetivo y el comando que se desea ejecutar (*OFF*).

```
public class ControladorInteraccionGUI {

    public static void procesar(String serial) {
        String[] datos = serial.split("-");
        String comando = datos[0];
        String dispObjetivo = datos[1];

        if (IdentificadorDispositivoObjetivo.consultarIsHabilitado(dispObjetivo)) {
            ManejadorControladorEjecucion.enviarSolicitudEjecutar(dispObjetivo, comando);
        }
    }
}
```

Figura 5.13: Implementación básica del Controlador de interacción GUI. Fuente: Elaboración propia.





```
public class ManejadorControladorEjecucion {

    public static void enviarSolicitudEjecutar(String dispObjetivo, String comando) {
        try {
            String mensaje = "/notificar?dispositivo=" + dispObjetivo + "&comando=" + comando;
            Notifier.printEnvioMensaje(CONTROLADOR, mensaje);
            HttpClientNode.enviar(mensaje);
        } catch (Exception ex) {
            Notifier.printInfoMensaje("El controlador de ejecución [" + CONTROLADOR + "/"
                                     + IP_CONTROLADOR + "] no está conectado: " + ex.getMessage());
        }
    }
}
```

Figura 5.14: Método que envía la solicitud de ejecución al Controlador de ejecución. Fuente: Elaboración propia.

#### 5.4.2.2. Controlador de interacción Gestual

Este controlador actúa como servicio externo al Middleware y se anexa a este mediante solicitudes y respuestas HTTP, esto con la finalidad de liberar al Middleware del procesamiento de señales gestuales y hacer que este proceso sea ejecutado por herramientas y algoritmos más eficientes. Recibe las imágenes captadas por una cámara de vídeo (Receptor de Interacción Gestual) correspondientes a las señales gestuales realizadas por el usuario, las procesa, interpreta y las envía al Middleware de interacción para la generación de la solicitud de ejecución, este proceso puede ser observado en los diagramas de secuencia mostrados en la Figura 5.3-5.4.

Las señales captadas por la cámara de video son gestos que el usuario hace con sus manos; se han definido dos tipos de gestos: uno para encender la bombilla de luz y el otro para apagarla. En la Figura 5.15 se muestra una representación del Repositorio de señales gestuales.


ID	GESTO	COMANDO	DISPOSITIVO OBJETIVO
G1		ON	FOCO1
G2		OFF	FOCO1

Figura 5.15: Representación del repositorio de señales gestuales. Fuente: Elaboración propia.

El algoritmo que ejecuta este Controlador de interacción para procesar las señales gestuales fue implementado en Python; además, hace uso de OpenCV<sup>2</sup>, que es una librería de visión artificial que cuenta con algoritmos para detectar y reconocer rostros, identificar objetos, rastrear movimientos de la cámara, etc.

En la Figura 5.16, se muestra el proceso de la detección de gestos con la mano. El algoritmo compara la imagen actual de la captura del video con una captura anterior y, como fondo, se definen los objetos que permanecen en la misma posición en las dos capturas. Esto genera una capa de primer plano que se traduce a una imagen binaria, la cual contiene regiones negras que son consideradas como el fondo y regiones blancas correspondientes al primer plano de la escena (mano).

El proceso es el siguiente: i) se obtiene la captura y se transforma a escala de grises, ii) luego se realiza un suavizado de la imagen (desenfoque) para reducir el ruido y iii) se lleva a cabo una binarización (Figura 5.16a). De la imagen binaria se buscarán los contornos, es decir, se dibuja una curva en la diferenciación del blanco y negro que formará el objeto que se encuentra en movimiento y se delimita por un rectángulo de área máxima (Figura 5.16b).

A continuación, se crea un envolvente convexo, que es un polígono convexo, el cual encierra el contorno detectado donde se crea los defectos de convexidad, que son las desviaciones máximas locales de la envoltura de los contornos (5.16c). Para encontrar el ángulo de todos los defectos (vértices) se aplica la regla del coseno, se ignora los ángulos mayores a 90° y, para los demás, se ejecuta un conteo, estos ángulos se muestran como puntos rojos en la Figura 5.16c, y su detección, mediante codificación Python, se muestra en la Figura 5.17.

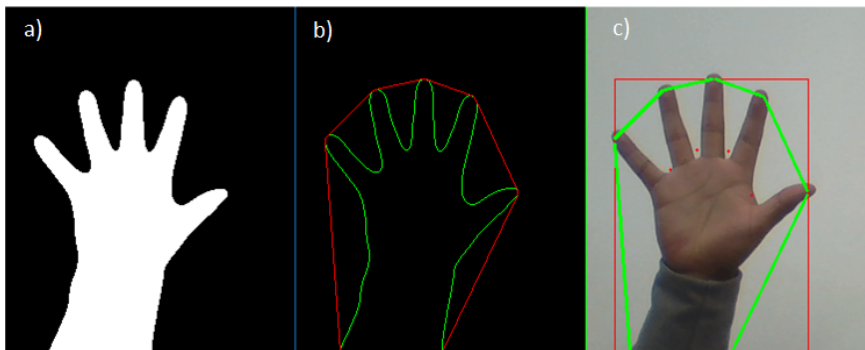


Figura 5.16: a) Binarización de la imagen. b) Dibujar contorno. c) Polígono convexo y vértices. Fuente: Elaboración propia.

<sup>2</sup>OpenCV, del inglés Open Source Computer Vision Library, en español Librería de Visión por Computadora de Código Abierto. Dirección Web: <https://opencv.org/>



```
for i in range(defects.shape[0]):
    s,e,f,d = defects[i,0]

    start = tuple(cnt[s][0])
    end = tuple(cnt[e][0])
    far = tuple(cnt[f][0])

    a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
    b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
    c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)

    angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57

    if angle <= 90:
        count_defects += 1
        cv2.circle(crop_img, far, 1, [0,0,255], -1)

    cv2.line(crop_img,start, end, [0,255,0], 2)
```

Figura 5.17: Código que aplica la regla de cosenos para contar los dedos mostrados en la mano (defectos o vértices). Fuente: Elaboración propia.

De esta manera, se puede identificar el número de dedos que están en la imagen. Entonces cuando se detecten cuatro defectos (mano abierta) se enviará el comando de “encender” al dispositivo (*ON*) o, cuando se detecte cero defectos (mano cerrada), se enviará el comando de “apagar” al dispositivo (*OFF*).

### 5.4.3. Módulo Controlador de ejecución

Este módulo consta del dispositivo que va a ser controlado por el usuario (bombilla de luz) y el Controlador de ejecución que va a accionar dicho dispositivo. En este caso, se ha optado por usar como Controlador de ejecución al módulo NodeMCU que incluye el chip WiFi ESP8266 y se empleó el lenguaje de programación Arduino para su configuración. La decisión de usar el módulo NodeMCU se fundamenta en el bajo coste que posee, su ahorro de energía, la posibilidad de programarlo usando el IDE Arduino desde una computadora, y debido a que el chip ESP8266 provee de una conexión WiFi a través de una pila TCP/IP completa y un microcontrolador.

#### 5.4.3.1. Controlador de ejecución

Respecto a las propiedades de conectividad dentro del entorno, el chip ESP8266 funciona como enrutador, es decir, el chip provee una infraestructura de red WiFi con acceso a Internet dentro del hogar y asigna direcciones

IP a todos los dispositivos con los que se ha establecido conexión. Esta configuración va a permitir que todos los dispositivos que el usuario puede controlar pertenezcan a una misma red inalámbrica, la cual está gestionada por el Controlador de ejecución; además, los dispositivos que no tengan la capacidad de acceder a una red inalámbrica, pueden ser controlados mediante una conexión eléctrica a los puertos de este componente.

La Figura 5.18 muestra una ilustración de las conexiones entre los dispositivos y el Controlador de ejecución.

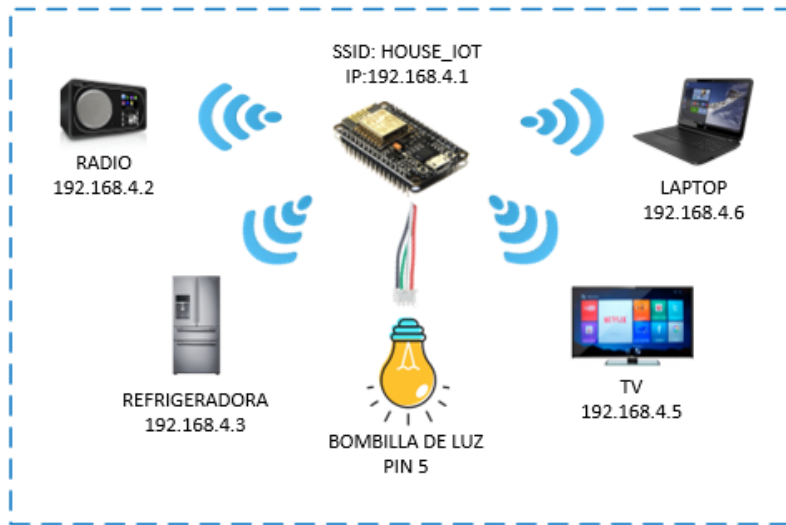


Figura 5.18: Ejemplo de conexión entre los Dispositivos Objetivo y el Controlador de ejecución. Fuente: Elaboración propia.

Para configurar las funcionalidades del Controlador de ejecución, se ha usado el lenguaje de programación Arduino. Este componente funciona como un servidor que expone diferentes servicios (Tabla 5.2) para la recepción de solicitudes HTTP desde el Middleware de Interacción.

La Figura 5.19 presenta el código cargado en el módulo NodeMCU para configurar las propiedades de conexión WiFi que tiene como enrutador y los servicios que brinda como servidor.



```
ESP8266WebServer server(80);

void setup() {
  pinMode(pinFocol,OUTPUT);
  digitalWrite(pinFocol,HIGH);
  Serial.begin(115200);
  setupWiFi();
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  server.on("/test",      handleTest);
  server.on("/notificar", handleNotificar);
  server.on("/estado",   handleEstado);
  server.on("/comandos", handleComandos);
  server.on("/dispositivos", handleDispositivos);
  server.begin();
  Serial.println("Servidor escuchando");
}

void loop() {
  server.handleClient();
}

void setupWiFi(){
  WiFi.mode(WIFI_AP);
  WiFi.softAP("HOUSE_IOT","1234asdf");
}
```

Figura 5.19: Configuración del módulo NodeMCU. Fuente: Elaboración propia.

A continuación, se brinda una descripción de las funciones más importantes que aparecen en la ilustración:

- *setup()*: esta es la función de configuración de los pines que se usarán para la conexión de los dispositivos que se desean controlar, se encarga de configurar la conexión a la red WiFi, define los servicios que serán expuestos por el servidor y las funciones que los manejan cuando son invocados y, finalmente, ejecuta el servidor.
- *loop()*: esta función se ejecuta como un hilo que permite que el servidor esté a la espera de clientes que invoquen sus servicios, cuando recibe una solicitud, se ejecuta el manejador correspondiente.
- *setupWiFi()*: esta función configura al chip ESP8266 como enrutador que provee conexiones WiFi a los dispositivos dentro del entorno; además, define el SSID (Identificador de Conjunto de Servicios, del inglés *Service Set Identifier*) de la red y la contraseña para la conexión.



La Figura 5.20 muestra la función que maneja la solicitud de ejecución de un comando sobre un dispositivo (función *handleNotificar()*), en este caso, el Middleware de interacción envía el comando y el dispositivo objetivo, el Controlador de ejecución recibe la solicitud, obtiene los parámetros e invoca a la función que ejecutará la acción solicitada (función *ejecutar()*).

```
void handleNotificar() {
    String message = "";
    String dispositivo = server.arg("dispositivo");
    String comando = server.arg("comando");
    if(dispositivo == "" || comando == ""){
        message = "ERROR";
    } else {
        message = "OK";
        ejecutar(dispositivo,comando);
    }

    server.send(200, "text/plain", message);
}

void ejecutar(String dispositivo, String comando){
    if(dispositivo == "FOCOL"){
        if(comando == "ON"){
            digitalWrite(pinFocol,LOW);
            estadoFocol = 1;
        }
        else if(comando == "OFF"){
            digitalWrite(pinFocol,HIGH);
            estadoFocol = 0;
        }
    }
}
```

Figura 5.20: Funciones que manejan las solicitudes de ejecución. Fuente: Elaboración propia.

#### 5.4.3.2. Dispositivo objetivo

Para este caso particular, el dispositivo objetivo corresponde a una bombilla de luz de 60W conectada al cableado eléctrico del entorno y al módulo NodeMCU a través de un dispositivo *Relé* que funciona como interruptor. El *Relé* se acciona al recibir un impulso eléctrico enviado por el módulo NodeMCU y amplifica la potencia al punto necesario para encender la bombilla de luz e interrumpe el flujo eléctrico cuando se desea apagar la bombilla. Las conexiones realizadas entre el dispositivo objetivo y el Controlador de ejecución se

presentan en la Figura 5.21.

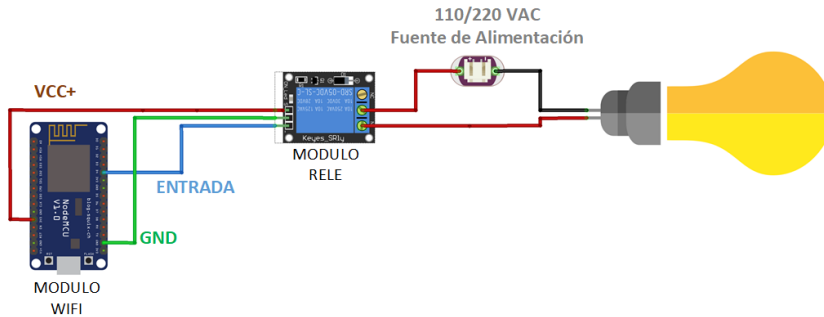


Figura 5.21: Conexión entre la bombilla de luz y el módulo NodeMCU. Fuente: Elaboración propia.

#### 5.4.3.3. Registro de Dispositivos Objetivo

La Figura 5.22 muestra el contenido de un fichero XML que almacena la información de los dispositivos presentes en el entorno sobre los cuales el Controlador de ejecución puede ejecutar los comandos enviados por el usuario. Este fichero contiene información relacionada al controlador (tipo, IP, puerto, dirección MAC), al entorno sobre el que actúa (tipo de entorno), y cada uno de los dispositivos objetivo (identificador, disponibilidad, tipo de dispositivo, IP o pin de conexión, estado actual, comandos que acepta).



```

1  <controlador>
2    <tipo>ESP8266</tipo>
3    <ip>192.168.4.1</ip>
4    <port>80</port>
5    <mac>22:34:12:45:56</mac>
6    <entorno>
7      <tipo>Habitacion</tipo>
8      <dispositivos>
9        <dispositivo id="FOCO1" habilitado="true">
10          <tipo>Bombilla de luz</tipo>
11          <estado>OFF</estado>
12          <comandos>
13            <comando>ON</comando>
14            <comando>OFF</comando>
15          </comandos>
16        </dispositivo>
17      </dispositivos>
18    </entorno>
19  </controlador>

```

Figura 5.22: Representación del Registro de dispositivos objetivo dentro del escenario planteado. Fuente: Elaboración propia.

## 5.5. Despliegue del prototipo del sistema

El prototipo del sistema fue desplegado en un ambiente de pruebas con usuarios reales los cuales pusieron a prueba el funcionamiento y constataron el cumplimiento de los requerimientos funcionales del sistema. Para la ejecución del prototipo, se siguió el siguiente procedimiento:

1. Se energizó el Controlador de ejecución (módulo NodeMCU) para que la red inalámbrica esté disponible en el entorno.
2. Se conectó el Dispositivo objetivo (bombilla de luz) al cableado eléctrico y al Controlador de ejecución.
3. Se ejecutó el Middleware de interacción en una computadora conectada a la red inalámbrica del entorno.
4. Se puso en marcha el Dispositivo receptor de interacción gestual (cámara de video) en otra computadora que también estaba conectada a la red inalámbrica del entorno.
5. Se puso en marcha el Dispositivo receptor de interacción GUI (aplicación móvil) en un smartphone con sistema operativo Android conectado a la red inalámbrica del entorno.
6. Finalmente, los usuarios empezaron a realizar las señales gestuales y las acciones en la GUI para encender y apagar la bombilla de luz.



La Figura 5.23 muestra a un usuario encendiendo y apagando la bombilla de luz usando los gestos definidos para la interacción gestual. Por otro lado, en la Figura 5.24 se puede observar a un usuario encendiendo y apagando la bombilla de luz usando los botones presentes en la interfaz gráfica de usuario.

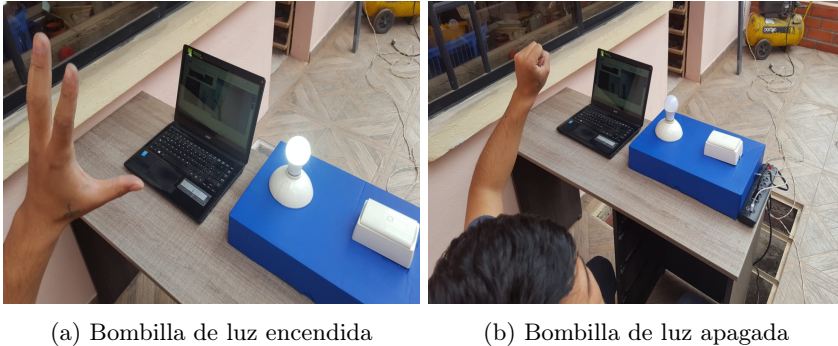


Figura 5.23: Usuario controlando la bombilla de luz mediante gestos. Fuente: Elaboración propia.

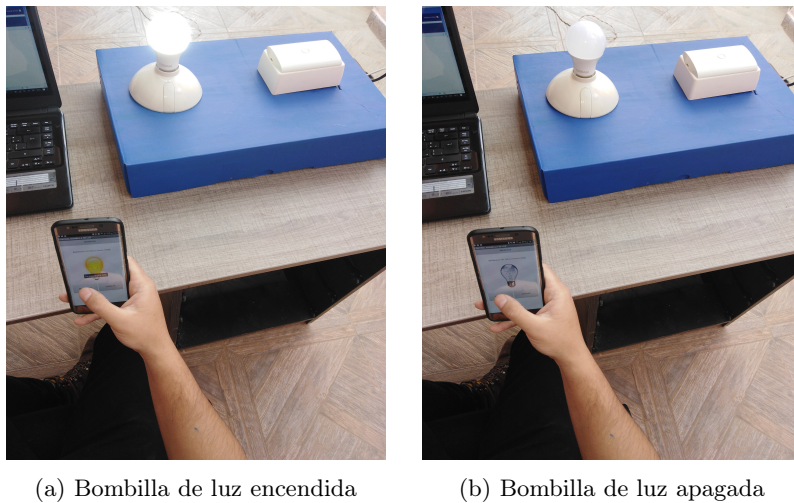


Figura 5.24: Usuario controlando la bombilla de luz mediante una GUI. Fuente: Elaboración propia.



## 5.6. Conclusiones

En este capítulo se ha presentado la implementación del prototipo de un sistema enfocado en la Interacción Hombre-Computadora (HCI) para Internet de las Cosas (IoT) dentro de Ambientes de Vida Asistidos (AAL). El sistema usa los módulos y componentes de la arquitectura de software AALIIoT de manera que puede ser visto como una instancia de dicha arquitectura. En este sentido, la arquitectura de software basada en HCI para IoT en AAL, brinda la estructura y los planos necesarios para la implementación de sistemas que fusionan las tres áreas: HCI, IoT y AAL. Esto se ha constatado durante el desarrollo del sistema para el escenario propuesto, en especial, durante el desarrollo de los componentes del Middleware de interacción, el Controlador de ejecución y la comunicación que se establece entre ellos, puesto que es la parte central de la arquitectura.

El escenario sobre el cual se basa la elicitación de requerimientos y el diseño e implementación del prototipo del sistema sobre el cual se desarrolla este capítulo, es un ejemplo típico de situaciones por las que tienen que atravesar las personas con movilidad reducida y adultos mayores en su vida diaria. En este caso, el prototipo del sistema implementado, satisface los requerimientos de un adulto mayor con discapacidad física, permitiéndole encender y apagar una bombilla de luz usando dos tipos de interacción: gestual y GUI.

Respecto a las herramientas bajo las cuales se pueden implementar los controladores de interacción, se considera adecuado emplear diferentes tecnologías, marcos de trabajo y APIs que permitan acelerar los tiempos de desarrollo y obtener resultados más eficaces al identificar comandos, mediante la reutilización de algoritmos probados y aceptados por la comunidad. Respecto a la implementación del Controlador de ejecución, se plantea como trabajo futuro la evaluación de los diferentes controladores que pueden ser usados en entornos específicos, por ejemplo: Raspberry Pi, Arduino UNO, Arduino Due, etc.





## Capítulo 6

# Evaluación empírica

En este capítulo se presenta la planificación y desarrollo de un cuasi-experimento cuyo objetivo es evaluar las percepciones de usuario acerca de la arquitectura de *software* AALIIoT; además, se evaluará cuán efectivo y eficiente puede ser un usuario al momento de realizar la selección de componentes necesarios para construir sistemas que consideren las necesidades de interacción de las personas, que estén basados en IoT y se desplieguen en entornos de vida asistidos.

En la sección 6.1 se presenta una introducción a la evaluación empírica de la solución propuesta, en la sección 6.2 se muestran los estudios empíricos empleados en el dominio de este trabajo. En la sección 6.3 se muestran los modelos teóricos de evaluación en Ingeniería de *Software*. La sección 6.4 presenta la adaptación del modelo de evaluación a ser utilizado para evaluar la arquitectura de *software*. La sección 6.5 muestra los cuasi-experimentos realizados sobre la arquitectura propuesta. La sección 6.6 presenta las amenazas a la validez y, finalmente, la sección 6.7 presenta las conclusiones de este capítulo.



## 6.1. Introducción

Las arquitecturas de *software* definen la estructura del sistema compuesta por diversos módulos, elementos de *software* y las relaciones que existen entre los diferentes componentes. El diseño adecuado de una arquitectura de *software*, garantiza en cierta medida que el sistema cumpla con uno o varios atributos de calidad, por ejemplo, que el sistema sea fácil de usar, confiable, seguro, eficiente, eficaz, etc [13]. En este sentido, es importante verificar que las arquitecturas de *software* cumplan con los atributos de calidad y que los usuarios perciban que son eficaces al momento de aplicarlas en la implementación de sistemas reales mediante estudios empíricos.

Los estudios empíricos, en el dominio de la Ingeniería del *Software*, sirven para evaluar los procesos, métodos, herramientas o cualquier otro tipo de tecnología creadas para el desarrollo, mantenimiento o aseguramiento de la calidad del *software* [12]. Un estudio empírico consiste en un acto u operación que permite descubrir algo que no se conoce o poner a prueba una hipótesis [11].

La aplicación del *Method Evaluation Model* (MEM) [56] es de gran utilidad a la hora de evaluar la eficacia percibida por los usuarios en la selección y adopción de una nueva solución. MEM fue originalmente propuesto como un medio para evaluar métodos de diseño de Sistemas de Información, sin embargo, ha conseguido ser exitosamente aplicado en la evaluación de otro tipo de métodos [3] [19]. Se eligió este modelo ya que integra variables del rendimiento de los usuarios y variables de su percepción de la solución como mecanismo para predecir la intención de uso y posible adopción en la práctica de la arquitectura de *software* AALIIoT. MEM extiende el *Technology Acceptance Model* (TAM) [24], el cual ha sido validado empíricamente en numerosos estudios que demuestran su utilidad para analizar la facilidad de uso percibida, utilidad percibida e intención de uso de los participantes para predecir su posible aceptación [19].

Para utilizar el MEM en la evaluación de arquitecturas de *software*, es necesario adaptar su modelo teórico para su uso en este dominio. Para ello, es necesario definir los constructores del modelo en función de las variables relevantes para arquitecturas de *software* y redefinir los ítems del cuestionario que permitirán medir las variables de percepción del usuario, su eficiencia y su eficacia al momento de dar soluciones a un conjunto de ejercicios planteados basados en escenarios específicos. Este capítulo presenta la adaptación realizada a MEM y tanto el diseño como la ejecución de cuasi-experimentos cuyos objetivos son proporcionar evidencias sobre la utilidad de la arquitectura de *software* AALIIoT.



## 6.2. Modelos teóricos de evaluación en Ingeniería de *Software*

Existen algunos modelos teóricos que permiten realizar validaciones en el dominio de la Ingeniería de *Software*. Específicamente, en el dominio de las arquitecturas de *software*, podemos encontrar métodos que se centran en analizar atributos de calidad específicos. Por ejemplo, el método *Architecture Level Modifiability Analysis* (ALMA), analiza la facilidad de modificación de una arquitectura debido a cambios en los requerimientos y a la necesidad de agregar nuevas funcionalidades [14]. Por otro lado, el método *Performance Assessment of Software Architecture* (PASA), analiza el desempeño o el tiempo que le toma a un sistema responder a ciertos eventos [83]. También se puede hacer referencia al método *Survivable Network Analysis* (SNA), que ayuda a identificar la capacidad de supervivencia de un sistema, analizando su arquitectura [50].

Es importante considerar el rol de las personas en el proceso de adopción de una solución a través de modelos teóricos que expliquen la aceptación de la tecnología, arquitecturas, metodologías y métodos. En este sentido, se puede decir que, si una arquitectura no es recordada como útil por los desarrolladores o ingenieros de software, sus expectativas de un despliegue exitoso serán muy limitadas, por ellos es importante medir la utilidad de las soluciones considerando la utilidad percibida, la facilidad de uso y la intención de uso futura.

El *Technology Acceptance Model* (TAM) presentado en [24], es un modelo teórico ampliamente usado dentro del campo de los Sistemas de Información, e intenta predecir y explicar el comportamiento en el uso de la tecnología; además, el *Method Evaluation Model* (MEM) propuesto por [56] provee mecanismos para evaluar el rendimiento actual, la aceptación y la posible adopción de un método de sistemas de información en la práctica. A continuación, se presentan los modelos teóricos más relevantes para el propósito de esta evaluación empírica.

### 6.2.1. *Technological acceptance model* (TAM)

TAM constituye una adaptación de la Teoría de Acción Razonada (TRA, por sus siglas en inglés, *Theory of Reasoned Action*) propuesta por [30]. TAM usa TRA como una base teórica para especificar los vínculos causales entre aspectos clave: utilidad percibida y facilidad de uso percibida con las actitudes, intenciones y comportamiento de los usuarios al momento de la adopción de una tecnología de computación. TAM es menos general que TRA, pero está diseñado exclusivamente para el comportamiento en el uso del computador y en la Ingeniería de *Software*.

De acuerdo a TRA, el rendimiento de una persona de una conducta específica está determinado por la intención de su comportamiento y la intención de su comportamiento está conjuntamente determinada por la actitud de la persona y las normas subjetivas concernientes al comportamiento en cuestión [24]. TAM utiliza dos adaptadores potenciales: la facilidad de uso percibida y la utilidad percibida de la tecnología como los principales determinantes de las actitudes hacia una nueva tecnología. Estas actitudes influyen las intenciones y de ahí el comportamiento. En este modelo, el uso es modelado como una función directa de la intención (Figura 6.1).

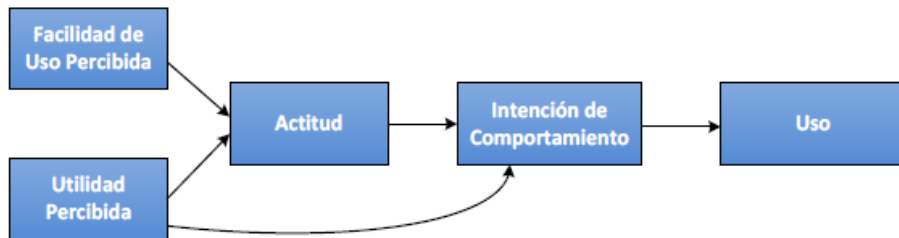


Figura 6.1: *Technology Acceptance Model* (TAM) simplificado. Fuente: [24].

El significado de cada constructor de TAM es:

- **Facilidad de Uso Percibida (PEOU, del inglés *Perceived Ease Of Use*):** el grado al cual los usuarios esperan que el sistema objetivo sea libre de esfuerzo.
- **Utilidad Percibida (PU, del inglés *Perceived Utility*):** la probabilidad subjetiva del usuario de que, utilizando una aplicación específica, podría incrementar su rendimiento laboral en un contexto organizacional.
- **Actitud (A):** el deseo del usuario para usar el sistema. Tanto PU como PEOU predicen la actitud hacia usar el sistema.
- **Intención de Comportamiento (IC):** la medida de la resistencia a ejecutar un comportamiento específico. A y PU influyen al individuo de IC a usar el sistema.
- **Uso:** el uso actual del sistema. Este es predicho por intenciones del comportamiento.

### 6.2.2. *Method Evaluation Model* (MEM)

La principal contribución de MEM consiste en extender el método TAM incorporando dos aspectos diferentes pero relacionados: la eficiencia actual y la efectividad actual. Esto significa que la adopción de una solución en la práctica, depende no solamente de si esta es efectiva (éxito pragmático) [3], sino además de si los usuarios de una solución la perciben efectiva (éxito percibido). En la Figura 6.2 se muestra el método MEM y los constructores del modelo, así como también las relaciones causales a lo largo de los constructores del mismo.

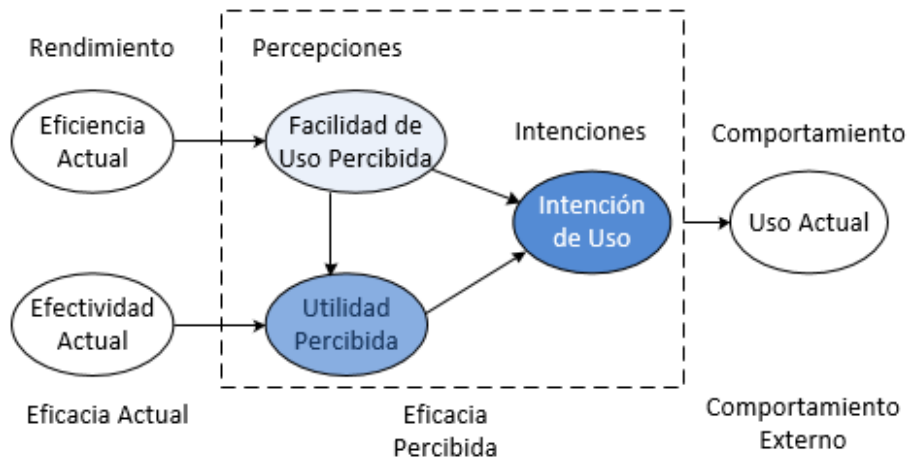


Figura 6.2: *Method Evaluation Model* – MEM. Fuente: [56].

En MEM, la eficacia es definida como un constructor separado, el cual es diferente de la eficiencia y la efectividad. El constructor de la eficacia está definido como la eficiencia y la efectividad con la cual un método consigue sus objetivos, es decir, para evaluar la eficacia se requiere la medición tanto del esfuerzo requerido (eficiencia) y la calidad de los resultados (efectividad) [69].

Considerando las definiciones dadas en [56], los constructores de MEM están basados en TAM, y son los siguientes:

- Eficacia Actual, este constructor tiene dos variables basadas en el rendimiento del usuario:
  - *Eficiencia Actual*: es el esfuerzo requerido para aplicar un método.
  - *Efectividad Actual*: es el grado en el cual un método consigue sus objetivos. Está relacionado a la calidad de los artefactos obtenidos





al momento de aplicar el método. Según [69], los métodos buscan conseguir ciertos objetivos mediante acciones, en este caso, la acción hace referencia al diseño de un sistema específico usando los componentes de la arquitectura propuesta.

- Eficacia Percibida, la cual tiene dos variables basadas en la percepción:
  - *Facilidad de Uso Percibida*: es el grado en el cual una persona cree que usando un método en particular puede estar libre de esfuerzo.
  - *Utilidad Percibida*: es el grado en el cual una persona cree que usando un método particular podría mejorar su rendimiento en el trabajo. Existe una relación causal en el modelo el cual indica que la utilidad percibida puede estar determinada por la facilidad de uso percibida.
  - *Intención de Uso*: es el modo en el que una persona intenta usar un método particular. Esta variable es usada para medir la probabilidad del método para ser adoptado en la práctica. Las relaciones causales sugeridas que perciben la facilidad de uso y la utilidad percibida directamente afectan la intención de usar el método.
- Uso Actual, el cual representa una variable basada en el comportamiento, definida como la intención de utilizar un método en la práctica. De acuerdo a la relación causal dada en MEM, el uso actual debe estar determinado por la intención del uso.

### 6.3. Adaptando el MEM para su uso en arquitecturas de *Software*

Para adaptar MEM en el uso de arquitecturas de software, en primera instancia, se deben definir los aspectos específicos de la arquitectura de software que serán evaluados; en base a estos aspectos, los constructores generales del MEM pueden ser instanciados en variables dependientes. Esta evaluación empírica se centra en evaluar la eficiencia, efectividad y percepciones del usuario al realizar la configuración de los componentes y módulos de la arquitectura propuesta para la creación de sistemas enfocados en HCI para aplicaciones de IoT en AAL.

La evaluación de la eficacia en la arquitectura involucra la medición del esfuerzo requerido para crear un sistema usando la arquitectura dado un escenario específico y la calidad de los resultados en la creación del sistema empleando la arquitectura. El esfuerzo requerido para entender y/o aplicar la



arquitectura (eficiencia actual) en la creación de un sistema pueden ser medidos mediante el tiempo o el esfuerzo cognitivo. La calidad del resultado de la arquitectura (efectividad actual) puede ser medida evaluando los resultados de la creación de un sistema haciendo uso de los componentes de la arquitectura. Dichas variables de rendimiento se emplean para medir la eficiencia y la efectividad de los usuarios utilizando la arquitectura para crear un sistema dado un escenario definidos y serán evaluadas de la siguiente manera:

- *Efectividad actual*: Se asignará una calificación según el grado de completitud, es decir en qué grado cada componente ha sido configurado (seleccionado y relacionado) correctamente en cada módulo.
  - 1, cuando los componentes han sido configurados correctamente.
  - 0.5, cuando los componentes han sido configurados parcialmente.
  - 0, cuando los componentes han sido configurados incorrectamente.

La efectividad total se evaluará con la ecuación 6.1, donde  $m$  representa a la cantidad de ejercicios propuestos y  $n$  la cantidad de tareas dentro de un ejercicio.

$$Efectividad = \frac{\sum_{j=1}^m (\sum_{i=1}^n calificacion\_tarea_{ij})}{m * n} \quad (6.1)$$

- *Eficiencia actual*: se mide sumando los tiempos empleados en la realización de cada tarea ejecutada por los participantes, para ello se usa la ecuación 6.2, donde  $m$  representa a la cantidad de ejercicios propuestos y  $n$  a la cantidad de tareas dentro de un ejercicio.

$$Eficiencia = \sum_{j=1}^m \left( \sum_{i=1}^n tiempo\_tarea_{ij} \right) \quad (6.2)$$

Se ha realizado una adaptación de un instrumento de medida utilizada en MEM, con la finalidad de medir variables basadas en la percepción. MEM ha sido operacionalizada para la creación de sistemas haciendo uso de la arquitectura propuesta, como se puede ver en la Figura 6.2, y así poder medir la facilidad de uso percibida (PEOU), utilidad percibida (PU) e intención de uso (ITU). Estos constructores han sido definidos mediante un conjunto de preguntas las que se describen en la Tabla 6.1. En la Figura 6.3 se puede visualizar el modelo teórico propuesto para evaluar la arquitectura.

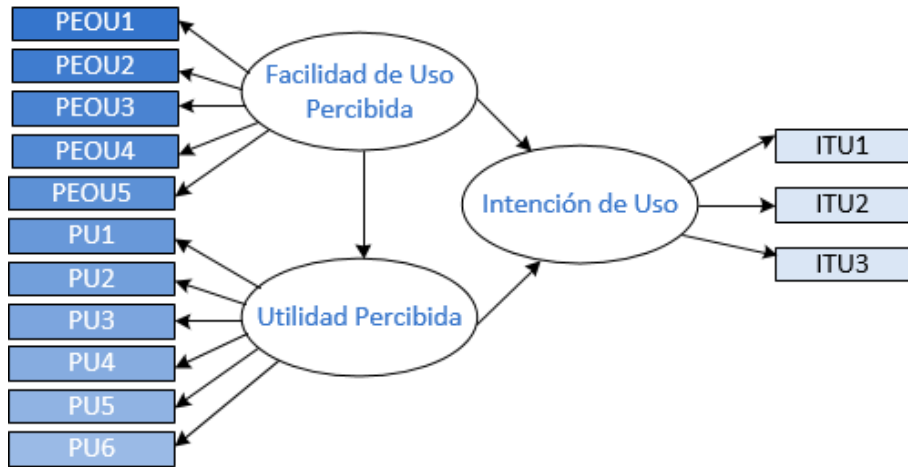


Figura 6.3: Distribución de preguntas del cuestionario. Fuente: [19].

Se han usado medidas basadas en el rendimiento como factores que influencian las variables de percepción, cada una de estas influencias serán probadas mediante la evaluación de diversas hipótesis, como se observa en la Figura 6.4.

La probabilidad de que la arquitectura sea aceptada en la práctica se puede analizar probando las siguientes hipótesis:

- $H1_0$ : La arquitectura se percibe como difícil de usar,  $H1_0 = \neg H1_1$
- $H2_0$ : La arquitectura no se percibe como útil,  $H2_0 = \neg H2_1$
- $H3_0$ : No existe intención de utilizar la arquitectura en el futuro,  $H3_0 = \neg H3_1$

Además, se plantean hipótesis que muestran una relación causal directa entre el uso de la arquitectura en un escenario particular y el rendimiento, percepciones e intenciones de los usuarios:

- $H4_0$ : La facilidad de uso percibida no puede verse determinada por la eficiencia,  $H4_0 = \neg H4_1$ , debido a que la eficiencia se mide en base a la eficiencia actual y la facilidad de uso en base de la percepción.
- $H5_0$ : La percepción de la utilidad no está determinada por la efectividad,  $H5_0 = \neg H5_1$ , debido a que la efectividad se mide en base al rendimiento y la utilidad percibida en base a la percepción de la efectividad.

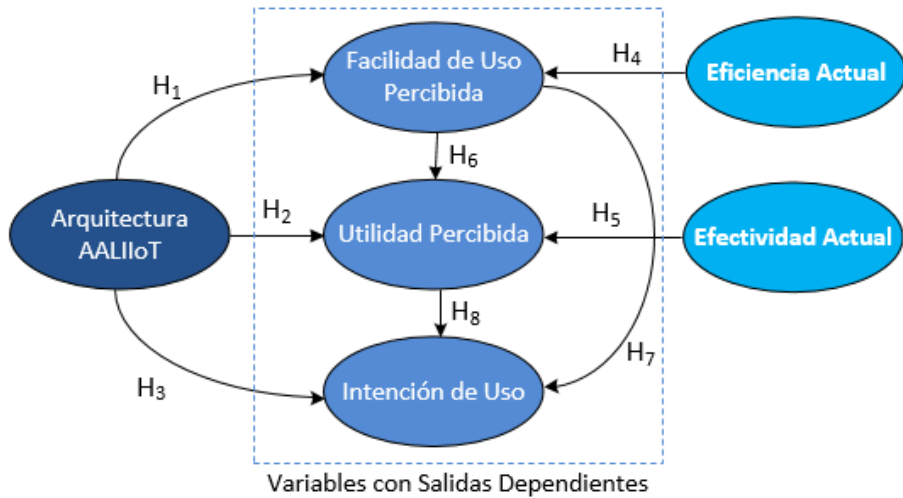


Figura 6.4: Modelo teórico para la evaluación de la arquitectura AALIIoT. Fuente: [19].

- $H_{60}$ : La utilidad percibida no es determinada por la facilidad de uso percibida,  $H_{60} = \neg H_{61}$ . Esta hipótesis es tomada desde el TAM, en el cual la facilidad de uso percibida se encuentra que no tiene una influencia directa sobre la utilidad percibida.
- $H_{70}$ : La intención de uso no es determinada por la facilidad de uso percibida,  $H_{70} = \neg H_{71}$ . Esta hipótesis es tomada desde el TAM en el cual se encuentra que la facilidad de uso percibida tiene influencia sobre la intención de uso.
- $H_{80}$ : La intención de uso no está determinada por la utilidad percibida.,  $H_{80} = \neg H_{81}$ . Esta hipótesis es tomada del TAM, en la cual se encontró que la utilidad percibida tiene una influencia directa sobre la intención de uso.

En la Tabla 6.1 se listan las preguntas definidas para medir las variables basadas en la percepción. Las preguntas fueron combinadas en un cuestionario con 14 preguntas. Son preguntas cerradas las cuales deben ser respondidas usando una escala de Likert del 1 al 5. La facilidad de uso percibida (PEOU) se mide utilizando cinco preguntas del cuestionario, la utilidad percibida (PU) se mide con seis preguntas y para la intención de uso futuro (ITU) se utilizó 3 preguntas. Con el fin de balancear las preguntas, algunas fueron negadas, para



evitar respuestas repetitivas. El cuestionario se puede encontrar en el Anexo E.2.

ID	PREGUNTA
PEOU1	La arquitectura es sencilla y fácil de seguir.
PEOU2	En general, la arquitectura es fácil de entender.
PEOU3	Los pasos a seguir para crear la arquitectura son claros y fáciles de entender.
PEOU4	La arquitectura es fácil de aprender.
PEOU5	Considero que sería fácil adquirir destrezas en el uso de la arquitectura.
PU1	Considero que la arquitectura reduciría el tiempo y el esfuerzo requerido para crear sistemas de este ámbito.
PU2	En general, considero que la arquitectura es útil.
PU3	Considero que la forma en que los componentes interactúan entre sí en la arquitectura es útil para la creación de sistemas de este ámbito.
PU4	Considero que la arquitectura es lo suficientemente expresiva para definir cómo los componentes interactúan entre sí.
PU5	El uso de esta arquitectura mejoraría mi rendimiento en la creación de sistemas de este ámbito.
PU6	En general, pienso que esta arquitectura permite cubrir adecuadamente los requisitos para sistemas de este ámbito.
ITU1	En caso de tener la necesidad de crear sistemas de este ámbito en el futuro, tendría en cuenta esta arquitectura.
ITU2	De ser necesario, utilizaría esta arquitectura.
ITU3	Recomendaría el uso de esta arquitectura.

Tabla 6.1: Cuestionario para medir variables de percepción.

Además, se incluyeron dos preguntas abiertas en el cuestionario sobre sugerencias y comentarios, que brindan una retroalimentación para mejorar la arquitectura en los aspectos que allí se mencionen. Las preguntas abiertas se muestran en la Tabla 6.2.



ID	PREGUNTA
PA1	¿Tiene alguna sugerencia de cómo hacer que esta arquitectura sea más fácil de usar?
PA2	¿Cuáles son las razones por las que tiene o no la intención de usar esta arquitectura en un futuro?

Tabla 6.2: Preguntas abiertas del cuestionario.

## 6.4. Evaluando la utilidad percibida de la arquitectura de software AALIIoT en la práctica mediante cuasi-experimentos

En esta sección, se presenta el experimento llevado a cabo para validar empíricamente la utilidad percibida por los usuarios sobre la arquitectura de software AALIIoT. Actualmente no existe, o no se ha encontrado, una arquitectura enfocada en la Interacción Hombre-Computador para aplicaciones de Internet de las Cosas en Ambientes de Vida Asistidos por lo que no se puede evaluar la arquitectura propuesta en este trabajo de titulación con respecto a otra arquitectura que sirva de control. Es por ello que se ha decidido llevar a cabo un cuasi-experimento con el objetivo de evaluar de forma empírica esta arquitectura.

El cuasi-experimento fue diseñado de acuerdo al proceso experimental propuesto en [84], en donde, además, se define a un cuasi-experimento como “una investigación empírica, similar a un experimento, en la cual la asignación de tratamiento a sujetos no puede estar basada en la aleatoriedad, pero emerge desde las características de los sujetos u objetos en sí mismo”. En este sentido, la arquitectura de software AALIIoT fue usada para predecir la probabilidad de aceptación de la actividad del diseño de sistemas en base a los componentes que forman parte de dicha arquitectura, considerando un estudio basado en las percepciones de los usuarios.

De acuerdo al paradigma GQM (del inglés, *Goal-Question Metric*) propuesta en [10] la meta de este experimento ha sido definida de la siguiente manera:

- **Evaluar:** el proceso de diseño de sistemas haciendo uso de los componentes de la arquitectura AALIIoT.
- **Con el propósito de:** evaluar la arquitectura propuesta con respecto a su eficacia.
- **Desde el punto de vista del:** arquitecto de software.



- **En el contexto de:** un grupo de alumnos, egresados y profesionales en el área de Ingeniería de Sistemas.

Las preguntas de investigación son:

- **RQ1:** ¿La arquitectura de software AALIIoT es percibida como fácil de usar y útil? De ser así, ¿las percepciones de los usuarios son el resultado de su rendimiento cuando utilizan la arquitectura para diseñar sistemas o aplicaciones enfocadas en la Interacción Hombre-Computador para aplicaciones de Internet de las Cosas en Ambientes de Vida Asistidos?
- **RQ2:** ¿Existe una intención de uso de la arquitectura de software AALIIoT en el futuro? De ser así, ¿tales intenciones de uso son el resultado de las percepciones de los participantes?

La primera pregunta de investigación puede ser estudiada mediante las siguientes hipótesis:

- $H1_0$ : La arquitectura se percibe como difícil de usar,  $H1_0 = \neg H1_1$
- $H2_0$ : La arquitectura no se percibe como útil,  $H2_0 = \neg H2_1$
- $H4_0$ : La facilidad de uso percibida no puede verse determinada por la eficiencia,  $H4_0 = \neg H4_1$
- $H5_0$ : La percepción de la utilidad no está determinada por la efectividad,  $H5_0 = \neg H5_1$

La segunda pregunta de investigación puede ser estudiada a través de las siguientes hipótesis:

- $H3_0$ : No existe intención de utilizar la arquitectura en el futuro,  $H3_0 = \neg H3_1$
- $H6_0$ : La utilidad percibida no es determinada por la facilidad de uso percibida,  $H6_0 = \neg H6_1$
- $H7_0$ : La intención de uso no es determinada por la facilidad de uso percibida,  $H7_0 = \neg H7_1$
- $H8_0$ : La intención de uso no está determinada por la utilidad percibida.,  $H8_0 = \neg H8_1$



## 6.4.1. Planificación del cuasi-experimento

### 6.4.1.1. Selección del contexto

El contexto está determinado por la arquitectura de software que será evaluada y la selección de los participantes. La arquitectura de software es la que se propone en este trabajo. La actividad consiste en diseñar sistemas que usen los componentes adecuados de la arquitectura para solventar los requerimientos de usuarios en escenarios particulares.

Esta actividad es importante debido a la interacción que tiene lugar entre el Arquitecto de software con los componentes y módulos de la arquitectura. Los sujetos ejecutarán el rol de diseñadores y analistas de sistemas basados en HCI para IoT en AAL ya que estarán a cargo de las siguientes actividades: (i) selección de dispositivos receptores de interacción en base a las necesidades de interacción de adultos mayores y personas con discapacidad, (ii) selección de controladores de interacción para la interpretación de comandos, (iii) selección de controladores de ejecución y dispositivos objetivos, y (iv) identificación de usuarios dentro del entorno AAL.

Finalmente, 30 participantes fueron seleccionados, de estos 18 son estudiantes de último año de la carrera de Ingeniería de Sistemas que han tomado cursos de Calidad de Software e Ingeniería de Software con conocimientos sobre modelos de calidad, métricas y arquitecturas de software. Además, 7 son estudiantes egresados de la carrera de Ingeniería de Sistemas y 5 son profesionales en el área de Ingeniería de Sistemas. Tanto egresados como profesionales, actualmente, desarrollan sus labores en análisis, diseño y programación de sistemas. Todos los participantes son, o fueron, estudiantes de la Universidad de Cuenca.

### 6.4.1.2. Tareas experimentales

El cuasi-experimento consiste en la resolución de tres ejercicios, los ejercicios se ambientan en escenarios específicos, cada ejercicio posee cuatro tareas; a continuación, se describe cada una:

- **Tarea 1:** Selección de dispositivos receptores de interacción. A partir de las necesidades de interacción del usuario identificadas dentro del escenario propuesto, el sujeto deberá elegir cuáles serán los dispositivos, dentro del Módulo Receptor de Interacción, que captarán los comandos (gestuales, vocales, mentales, etc.) del usuario cuando este quiera realizar alguna acción sobre algún dispositivo que desee controlar, por ejemplo, mediante un micrófono, una cámara, un casco para BCI, etc.





- **Tarea 2:** Selección de controladores de interacción e identificador de dispositivos objetivos. Nuevamente, a partir de las necesidades de interacción del usuario identificadas dentro del escenario propuesto, el sujeto deberá elegir cuáles serán los controladores de interacción que el Middleware de interacción debe poseer para recibir y procesar las señales enviadas por los dispositivos receptores de interacción seleccionados en la Tarea 1. Además, el sujeto debe indicar el uso del identificador de dispositivos sobre los cuales se van a ejecutar los comandos del usuario. Finalmente, establecer las conexiones que tendrán los componentes elegidos hasta este punto.
- **Tarea 3:** Selección de controladores de ejecución y dispositivos objetivos. El sujeto deberá seleccionar, para el Módulo Controlador de ejecución, el controlador que considere adecuado para enviar las señales de ejecución respectivas a los dispositivos que el usuario desea controlar, de igual forma, estos dispositivos también deberán ser seleccionados por el sujeto y, además, deberá agregarlos a un registro. Finalmente, establecerá las conexiones que tendrán los componentes elegidos hasta este punto.
- **Tarea 4:** Selección del tipo de identificación de usuarios y los dispositivos necesarios para llevarla a cabo. Si se deduce que, en el escenario planteado, es necesario identificar a los usuarios antes de ejecutar las acciones que estos requieran llevar a cabo, entonces el sujeto deberá seleccionar el tipo de identificación que considere adecuada para el Módulo de Identificación de Usuarios y los dispositivos receptores de interacción que serán los encargados de tomar la información necesaria del usuario para proceder con su identificación. Finalmente, establecerá las conexiones que tendrán los componentes elegidos hasta este punto.

#### 6.4.1.3. Variables

En la Tabla 6.3 se presentan las variables dependientes de interés basadas en la percepción, de acuerdo al MEM, las cuales fueron usadas para evaluar la arquitectura propuesta en la práctica.

Estas variables son medidas usando un cuestionario con una escala de Likert con un conjunto de 14 preguntas cerradas (5 para facilidad de uso percibida, 6 para utilidad percibida y 3 para intención de uso futura). Las preguntas cerradas fueron formuladas utilizando una escala de Likert de 5 puntos. El valor agregado para cada variable subjetiva fue calculado como la media aritmética de las respuestas a las preguntas asociadas con cada variable subjetiva dependiente.



VARIABLE	DESCRIPCIÓN
Facilidad de Uso Percibida (PEOU)	El grado en el cual los participantes creen que al aprender y usar la arquitectura propuesta estarán libres de esfuerzo.
Utilidad Percibida (PU)	El grado en el cual los participantes creen que usando arquitectura propuesta se incrementará su rendimiento.
Intención de Uso (ITU)	El grado en el cual los participantes piensan usar la arquitectura propuesta de necesitar una arquitectura basada en HCI para IoT dentro de AAL en el futuro. Esto representa un juicio de la eficacia del método y puede ser utilizado para predecir la aceptación del método en la práctica.

Tabla 6.3: Variables dependientes basadas en la percepción.

En la Tabla 6.4 se muestran las variables basadas en el rendimiento de interés y la función de medición usada para determinar sus valores.

VARIABLE	ECUACIÓN
Efectividad	$Efectividad = \frac{\sum_{j=1}^m (\sum_{i=1}^n calificacion\_tarea_{ij})}{m * n}$
Eficiencia	$Eficiencia = \sum_{j=1}^m \left( \sum_{i=1}^n tiempo\_tarea_{ij} \right)$

Tabla 6.4: Variables dependientes basadas en el rendimiento.

#### 6.4.1.4. Material experimental

El material experimental se compone del conjunto de documentos que son necesarios para realizar las tareas experimentales y el cuestionario para medir la percepción del usuario una vez que se ha realizado el experimento. Este material incluye todo lo referente a las presentaciones con los conceptos, explicación de tareas y ejemplos que son utilizadas para el entrenamiento de los participantes.

La documentación utilizada, en su totalidad, ha sido incluida en el Apéndice



E de este trabajo de titulación y es explicada a continuación:

1. Un folleto que contiene la descripción del proceso de evaluación, un ejemplo específico ilustrando la forma de desarrollar las cuatro tareas y tres ejercicios propuestos para que sean ejecutados por los participantes. Se solicitó a los participantes que escriban la hora exacta de inicio y fin de cada una de las tareas a resolver.
2. Un anexo detallado como soporte, el cual describe cada módulo y componente de la arquitectura.
3. Un anexo con una ilustración de la arquitectura propuesta.
4. Un anexo con una guía del proceso de ejecución de la evaluación para ser utilizada durante el experimento como material de referencia.
5. Tres hojas de resolución para que el participante plasme sus respuestas. Una hoja para cada ejercicio.
6. Un cuestionario que contiene preguntas cerradas para analizar las variables subjetivas y algunas respuestas abiertas para permitir a los participantes expresar su opinión sobre la arquitectura.

El tiempo gastado en cada tarea y los datos obtenidos como resultado fueron recolectados utilizando las hojas de resolución descritas en el numeral cinco. El cuestionario fue recolectado en línea. Todos los documentos fueron creados en español, ya que éste es el idioma nativo de los participantes en todos los experimentos.

#### **6.4.2. Operación y ejecución del cuasi-experimento**

Se realizaron dos evaluaciones individuales, para cada una de ellas se efectuó una sesión de entrenamiento de 60 minutos antes de la sesión experimental, considerando la homogeneidad de los participantes, con la finalidad de presentar los conceptos relacionados a la arquitectura de software AALIoT, explicar el material experimental y desarrollar un ejemplo demostrativo. La Figura 6.5 brinda un resumen de los cuasi-experimentos realizados.

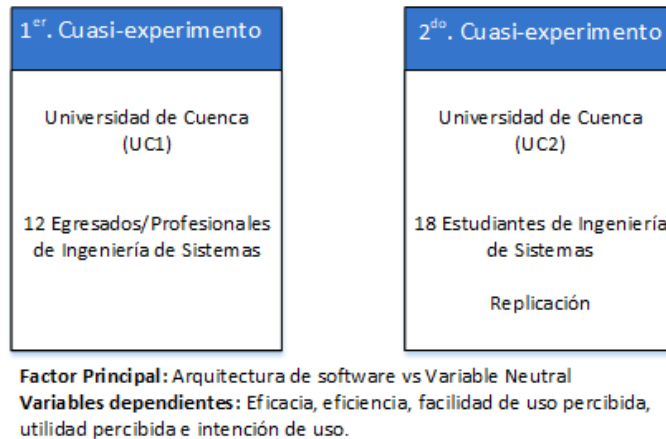


Figura 6.5: Resumen de los cuasi-experimentos realizados. Fuente: Elaboración propia.

En el primer cuasi-experimento (egresados y profesionales) participaron 12 sujetos que actualmente están encaminados al desarrollo de sus actividades en el ámbito profesional, todos ellos fueron estudiantes de la carrera de Ingeniería de Sistemas de la Universidad de Cuenca. El segundo cuasi-experimento (estudiantes) corresponde a una réplica del primer cuasi-experimento y se realizó sobre un grupo de 18 estudiantes que actualmente cursan el último año de la carrera de Ingeniería de Sistemas de la Universidad de Cuenca.

### 6.4.3. Ejecución y análisis de los cuasi-experimentos individuales

En esta sección se describen las características de cada uno de los cuasi-experimentos realizados. Para realizar los análisis y establecer criterios de aceptación o rechazo de hipótesis, se han utilizado los niveles de significancia sugeridos en [56] los cuales se muestra en la Tabla 6.5.

VALOR DE SIGNIFICANCIA	RANGO
No significativo	$p > 0.1$
Baja significancia	$p < 0.1$
Media significancia	$p < 0.05$
Alta significancia	$p < 0.01$
Muy alta significancia	$p < 0.001$



Tabla 6.5: Niveles de significancia [56].

#### 6.4.3.1. cuasi-experimento UC1 (Egresados y Profesionales)

Este cuasi-experimento fue realizado en la Universidad de Cuenca con un total de 12 participantes en una sesión de 60 minutos de duración de los cuales 20 minutos fueron de entrenamiento y 40 para la ejecución del cuasi-experimento. Los evaluadores clarificaron cualquier duda o pregunta a lo largo de la sesión. Al finalizar, los participantes llenaron el cuestionario post-experimento. Para realizar el análisis de resultados, se usaron pruebas, estadística descriptiva y *box plots* sobre los datos recolectados. Los resultados fueron obtenidos usando SPSS v20 con un  $\alpha = 0.05$ .

#### Análisis de las percepciones de usuario

La Figura 6.6 muestra los diagramas de caja para cada variable de percepción en donde se observa que la media de cada variable es mayor que 3, que representa al valor neutro de la escala de Likert.

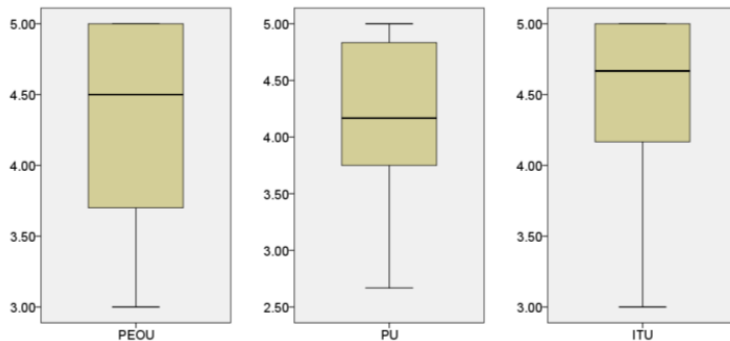


Figura 6.6: UC1 - Diagrama de cajas para las variables PEOU, PU e ITU. Fuente: Elaboración propia.

A continuación, se realizó una prueba de normalidad para comprobar si los datos poseen una distribución paramétrica o no y así determinar qué pruebas podrían usarse para verificar las hipótesis H1, H2 y H3. Debido a que el cuasi-experimento se realizó con 12 sujetos, se aplicó la prueba de Shapiro-Wilk.

Para las variables PEOU e ITU, las cuales tienen una distribución no paramétrica (\*\*) ( $p < 0.05$ ), se probó la hipótesis aplicando el *test Wilcoxon one-*



*tailed one sample* con un valor de prueba igual a 3, debido a que este valor corresponde al valor neutral de la escala de Likert del cuestionario, mientras que para la variable PU, que sí tiene distribución paramétrica (\*) ( $p > 0.05$ ), se aplicó el *t-test one-tailed*. Estos resultados, junto con los valores de la media, y los test de normalidad Shapiro-Wilk son presentados en la Tabla 6.6.

Var	Min	Max	Media	Std. Dev	Std E.	1-T. p-value	Shapiro- Wilk test p-value
PEOU	3.00	5.00	4.35	0.75859	0.21899	0.003 <0.01**	0.014 <0.05
PU	2.67	5.00	4.1667	0.73168	0.21122	0.000 <0.01*	0.324 >0.05
ITU	3.00	5.00	4.50	0.59459	0.17164	0.003 <0.01**	0.01 <0.05

Tabla 6.6: UC1 - Resultados del análisis de percepciones del usuario.

De los resultados presentados en la Tabla 6.6, se puede observar que los datos obtenidos al aplicar las pruebas *t-test one-tailed* y el *test* de Wilcoxon (columna “1-T. p-value”), son menores que 0.01 ( $p < 0.01$ ) lo cual, según los niveles de significancia mostrados en la Tabla 6.5, indica que poseen un alto grado de significancia. Esto permite rechazar las hipótesis nulas  $H1_0$ ,  $H2_0$  y  $H3_0$ , lo que significa que la arquitectura de software AALIIoT es percibida como fácil de usar, útil y que los participantes consideran usarla en el futuro cuando tengan que diseñar sistemas basados en HCI para IoT dentro de AAL.

## Análisis del Rendimiento del Usuario

Se realizó una medición de la efectividad y la eficiencia de los participantes al diseñar sistemas para escenarios específicos utilizando los módulos y componentes de la arquitectura de software AALIIoT en la práctica. La Tabla 6.7 muestra los valores de estadística descriptiva para las variables de efectividad y eficiencia. Los participantes tuvieron un promedio del 86.25 % de efectividad al completar los ejercicios experimentales.

La eficiencia se calculó como el esfuerzo requerido en minutos para resolver los ejercicios experimentales [56]. Los participantes tuvieron una eficiencia que fue de 8.25 a 15 minutos. Es necesario considerar que la eficiencia puede variar dependiendo de varios factores, como, por ejemplo, la experiencia de los sujetos en el dominio al cual pertenece la arquitectura.



Variable	Min	Max	Media	Std. Dev
Efectividad	0.71	1	0.8625	0.09498
Eficiencia	8.25	15	12.1458	2.12846

Tabla 6.7: UC1 - Estadística descriptiva para variables basadas en el Rendimiento del Usuario.

## Análisis de Relaciones Causales

En esta sección se realiza la validación de la parte estructural de MEM en términos de las relaciones causales que hay entre sus constructores, exceptuando el Uso Actual. Para ello, se usa un análisis de regresión que permita evaluar la operacionalización de MEM, debido a que las hipótesis son relaciones causales entre variables continuas. Para efectuar este análisis, se usan los niveles de significancia dados por [56] que fueron presentados en la Tabla 6.5.

## Eficiencia vs Facilidad de Uso Percibida

La hipótesis H4 se ha definido con el objetivo de comprobar si las percepciones de Facilidad de Uso Percibida (PEOU) son determinadas por la Eficiencia de los participantes al usar la arquitectura. El análisis consiste en la construcción de un modelo de regresión simple en donde la Eficiencia es la variable independiente y PEOU la variable dependiente. La ecuación de regresión resultante del análisis es la siguiente:

$$PEOU = 7,362 + (-0,248) * Eficiencia \quad (6.3)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	7.362	0.998		7.379	0.000		
Eficiencia	-0.248	0.081	-0.696	-3.061	0.012 <0.05	0.696	0.484

Tabla 6.8: UC1 - Regresión Simple entre la Eficiencia Actual y la Facilidad de Uso Percibida.

El modelo de regresión presenta una significancia media, con  $p < 0.05$ . El  $R^2$  muestra que la variable Eficiencia permite explicar el 48.4 % de la varianza de PEOU, lo cual indica que la eficiencia actual de los participantes influencia a ciertas percepciones de facilidad de uso. Estos resultados nos permiten rechazar la hipótesis nula  $H_{40}$  y aceptar su hipótesis alternativa, lo que significa que se



ha encontrado que la facilidad de uso percibida (PEOU) sí está determinada por la Eficiencia de los participantes.

### Efectividad vs Utilidad Percibida

La hipótesis H5 se ha definido con el objetivo de comprobar si las percepciones de la Utilidad Percibida (PU) están determinadas por la Efectividad de los participantes. De igual forma, se ha realizado la construcción de un modelo de regresión simple en donde la Efectividad es la variable independiente y PU la variable dependiente. La ecuación de regresión resultante del análisis es la siguiente:

$$PU = 3,194 + (1,129) * Efectividad \quad (6.4)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	3.194	2.096		1.524	0.158		
Efectividad	1.129	2.420	0.146	0.467	0.651 >0.1	0.146	0.021

Tabla 6.9: UC1 - Regresión Simple entre la Efectividad Actual y la Utilidad Percibida.

El modelo de regresión no es significativo, con  $p > 0.1$ . El  $R^2$  muestra que la variable Efectividad permite explicar solamente el 2.1 % de la varianza de PU, lo cual indica que muy pocas de las percepciones con respecto a PU están determinadas por la efectividad de los participantes. Estos resultados no nos permiten rechazar la hipótesis nula  $H5_0$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la utilidad percibida (PU) no está determinada por la Efectividad de los participantes.

### Facilidad de Uso Percibida vs Utilidad Percibida

La hipótesis H6 se ha definido con el objetivo de comprobar si las percepciones de la Utilidad Percibida (PU) están determinadas por la Facilidad de Uso Percibida (PEOU). De igual forma, se ha realizado la construcción de un modelo de regresión simple en donde PEOU es la variable independiente y PU la variable dependiente. La ecuación de regresión resultante del análisis es la siguiente:

$$PU = 0,754 + (0,785) * PEOU \quad (6.5)$$





Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	0.754	0.782		0.963	0.358		
PEOU	0.785	0.177	0.813	4.423	0.001 <0.01	0.813	0.662

Tabla 6.10: UC1 - Regresión Simple entre la Facilidad de Uso Percibida y la Utilidad Percibida.

El modelo de regresión es altamente significativo, con  $p < 0.01$ . El  $R^2$  muestra que la variable PEOU permite explicar el 66.2 % de la varianza de PU, lo cual indica que ciertas percepciones con respecto a PU están determinadas por PEOU. Estos resultados nos permiten rechazar la hipótesis nula  $H_0$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la utilidad percibida (PU) sí está determinada por la facilidad de uso percibida (PEOU) de los participantes.

### Intención de Uso vs Facilidad de Uso Percibida

La hipótesis H7 se ha definido con el objetivo de comprobar si las percepciones de la Intención de Uso (ITU) están determinadas por la Facilidad de Uso Percibida (PEOU). Para el análisis, se ha realizado la construcción de un modelo de regresión simple en donde PEOU es la variable independiente e ITU la variable dependiente. La ecuación de regresión resultante del análisis es la siguiente:

$$ITU = 1,866 + (0,606) * PEOU \quad (6.6)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	1.866	0.694		2.688	0.023		
PEOU	0.606	0.157	0.773	3.848	0.003 <0.01	0.773	0.597

Tabla 6.11: UC1 - Regresión Simple entre la Facilidad de Uso Percibida y la Intención de Uso.

El modelo de regresión es altamente significativo, con  $p < 0.01$ . El  $R^2$  muestra que la variable PEOU permite explicar el 59.7 % de la varianza de ITU, lo cual indica que existe un alto grado de que ITU esté determinada por las percepciones de facilidad de uso (PEOU). Estos resultados nos permiten rechazar la hipótesis nula  $H_0$  y aceptar su hipótesis alternativa, lo que significa



que se ha encontrado que la intención de uso (ITU) sí está determinada por la facilidad de uso percibida (PEOU) de los participantes.

### Intención de Uso vs Utilidad Percibida

La hipótesis H8 se ha definido con el objetivo de comprobar si las percepciones de la Intención de Uso (ITU) están determinadas por la Utilidad Percibida (PU). Para el análisis, se ha realizado la construcción de un modelo de regresión simple en donde PU es la variable independiente e ITU la variable dependiente. La ecuación de regresión resultante del análisis es la siguiente:

$$ITU = 1,827 + (0,642) * PU \quad (6.7)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	1.827	0.667		2.741	0.021		
PU	0.642	0.158	0.789	4.067	0.002 <0.01	0.789	0.623

Tabla 6.12: UC1 - Regresión Simple entre la Utilidad Percibida y la Intención de Uso.

El modelo de regresión es altamente significativo, con  $p < 0.01$ . El  $R^2$  muestra que la variable PU permite explicar el 62.3 % de la varianza de ITU, lo cual indica que existe un alto grado de que ITU esté determinada por las percepciones de utilidad (PU). Estos resultados nos permiten rechazar la hipótesis nula  $H8_0$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la intención de uso (ITU) sí está determinada por la utilidad percibida (PU) de los participantes.

#### 6.4.3.2. cuasi-experimento UC2 (Estudiantes)

Este cuasi-experimento fue realizado en la Universidad de Cuenca con un total de 18 participantes. El procedimiento fue el mismo que el usado durante la ejecución y análisis del cuasi-experimento original (egresados y profesionales).

#### Análisis de las percepciones de usuario

La Figura 6.7 muestra los diagramas de caja para cada variable de percepción en donde se observa que la media de cada variable es mayor que 3, que representa al valor neutro de la escala de Likert.

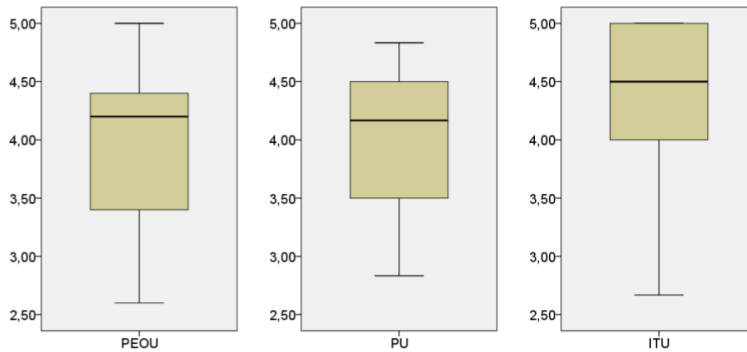


Figura 6.7: UC2 - Diagrama de cajas para las variables PEOU, PU e ITU.  
Fuente: Elaboración propia.

A continuación, similar al cuasi-experimento UC1 se realizó una prueba de normalidad para determinar los tests que verifiquen las hipótesis H1, H2 y H3. Debido a que el cuasi-experimento se realizó con 18 sujetos, se aplicó la prueba de Shapiro-Wilk.

Para la variable ITU, la cual tiene una distribución no paramétrica (\*\*) ( $p < 0.05$ ), se probó la hipótesis aplicando el test Wilcoxon one-tailed one sample con un valor de prueba igual a 3 (valor neutral), mientras que para las variables PEOU y PU, que si tienen distribución paramétrica (\*) ( $p > 0.05$ ), se aplicó el t-test one-tailed. Los resultados son presentados en la Tabla 6.13.

Var	Min	Max	Media	Std. Dev	Std. E.	1-T. p-value	Shapiro- Wilk test p-value
PEOU	2.6	5	3.9556	0.71144	0.16769	0.000 <0.001*	0.263 >0.05
PU	2.83	4.83	4.0556	0.59409	0.14003	0.000 <0.001*	0.137 >0.05
ITU	2.67	5	4.2963	0.7829	0.18453	0.000 <0.001**	0.006 <0.05

Tabla 6.13: UC2 - Resultados del análisis de percepciones del usuario.

De los resultados presentados en la Tabla 6.13, se puede observar que los datos obtenidos al aplicar las pruebas t-test one-tailed y el test de Wilcoxon (columna “1-T. p-value”), son menores que 0.001 ( $p < 0.001$ ) lo cual, según los niveles de significancia mostrados en la Tabla 6.5, indica que poseen un grado de significancia muy alto. Esto permite rechazar las hipótesis nulas  $H1_0$ ,  $H2_0$  y  $H3_0$ , lo que significa que la arquitectura de software AALIIoT es percibida como fácil de usar, útil y que los participantes consideran usarla en el futuro cuando tengan que diseñar sistemas basados en HCI para IoT dentro de AAL.



## Análisis del Rendimiento del Usuario

Se realizó una medición de la efectividad y la eficiencia de los participantes al diseñar sistemas para escenarios específicos utilizando los módulos y componentes de la arquitectura de software AALIoT en la práctica. La Tabla 6.14 muestra los valores de estadística descriptiva para las variables de efectividad y eficiencia. Los participantes tuvieron un promedio del 70.94 % de efectividad al completar los ejercicios experimentales.

En cuanto a la eficiencia, calculada como el esfuerzo requerido en minutos para resolver los ejercicios experimentales, a los participantes les tomó entre 7.88 y 35 minutos para completar las tareas. Se considera que, en este caso, la eficiencia y la efectividad se vieron afectadas principalmente debido a la falta de experiencia de los sujetos en el dominio al cual pertenece la arquitectura.

Variable	Min	Max	Media	Std. Dev
Efectividad	0.46	0.96	0.7094	0.14194
Eficiencia	7.88	35	17.405	6.44788

Tabla 6.14: UC2 - Estadística descriptiva para variables basadas en el Rendimiento del Usuario.

## Análisis de Relaciones Causales

En esta sección se realiza la validación de la parte estructural de MEM en términos de las relaciones causales que hay entre sus constructores, exceptuando el Uso Actual. Para ello, se usa un análisis de regresión que permita evaluar la operacionalización de MEM, debido a que las hipótesis son relaciones causales entre variables continuas. Para efectuar este análisis, se usan los niveles de significancia dados por [56] que fueron presentados en la Tabla 6.5.

### Eficiencia vs Facilidad de Uso Percibida

La hipótesis H4 comprueba si las percepciones de Facilidad de Uso Percibida (PEOU) son determinadas por la Eficiencia de los participantes al usar la arquitectura. Se construyó un modelo de regresión simple en donde la Eficiencia es la variable independiente y PEOU la variable dependiente. La ecuación de regresión resultante es la siguiente:

$$PEOU = 3,430 + (0,030) * Eficiencia \quad (6.8)$$



Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	3.430	0.491		6.989	0.000		
Eficiencia	0.030	0.027	0.273	1.137	0.272 >0.1	0.273	0.075

Tabla 6.15: UC2 - Regresión Simple entre la Eficiencia Actual y la Facilidad de Uso Percibida.

El modelo de regresión no es significativo, con  $p > 0.1$ . El  $R^2$  muestra que la variable Eficiencia permite solamente el 7.5 % de la varianza de PEOU, lo cual indica que la eficiencia actual de los participantes no influencia las percepciones de facilidad de uso. Estos resultados no nos permiten rechazar la hipótesis nula  $H_{40}$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la facilidad de uso percibida (PEOU) no está determinada por la Eficiencia de los participantes.

### Efectividad vs Utilidad Percibida

La hipótesis  $H_5$  comprueba si las percepciones de la Utilidad Percibida (PU) están determinadas por la Efectividad de los participantes. De igual forma, se ha realizado la construcción de un modelo de regresión simple en donde la Efectividad es la variable independiente y PU la variable dependiente. La ecuación de regresión resultante es la siguiente:

$$PU = 3,252 + (1,134) * Efectividad \quad (6.9)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	3.252	0.733		4.436	0.000		
Efectividad	1.134	1.016	0.269	1.116	0.281 >0.1	0.269	0.072

Tabla 6.16: UC2 - Regresión Simple entre la Efectividad Actual y la Utilidad Percibida.

El modelo de regresión no es significativo, con  $p > 0.1$ . El  $R^2$  muestra que la variable Efectividad permite explicar solamente el 7.2 % de la varianza de PU, lo cual indica que muy pocas de las percepciones con respecto a PU están determinadas por la efectividad de los participantes. Estos resultados no nos permiten rechazar la hipótesis nula  $H_{50}$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la utilidad percibida (PU) no está determinada por la Efectividad de los participantes.



### Facilidad de Uso Percibida vs Utilidad Percibida

La hipótesis H6 comprueba si las percepciones de la Utilidad Percibida (PU) están determinadas por la Facilidad de Uso Percibida (PEOU). De igual forma, se ha realizado la construcción de un modelo de regresión simple en donde PEOU es la variable independiente y PU la variable dependiente. La ecuación de regresión resultante es la siguiente:

$$PU = 1,139 + (0,737) * PEOU \quad (6.10)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	1.139	0.393		2.895	0.011		
PEOU	0.737	0.098	0.883	7.525	0.000 <0.001	0.883	0.780

Tabla 6.17: UC2 - Regresión Simple entre la Facilidad de Uso Percibida y la Utilidad Percibida.

El modelo de regresión posee una significancia muy alta, con  $p < 0.001$ . El  $R^2$  muestra que la variable PEOU permite explicar el 78 % de la varianza de PU, lo cual indica que la gran mayoría de percepciones con respecto a PU están determinadas por PEOU. Estos resultados nos permiten rechazar la hipótesis nula  $H_{60}$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la utilidad percibida (PU) sí está determinada por la facilidad de uso percibida (PEOU) de los participantes.

### Intención de Uso vs Facilidad de Uso Percibida

La hipótesis H7 comprueba si las percepciones de la Intención de Uso (ITU) están determinadas por la Facilidad de Uso Percibida (PEOU). Para el análisis, se ha realizado la construcción de un modelo de regresión simple en donde PEOU es la variable independiente e ITU la variable dependiente. La ecuación de regresión resultante es la siguiente:

$$ITU = 0,908 + (0,857) * PEOU \quad (6.11)$$



Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	0.908	0.694		1.309	0.209		
PEOU	0.857	0.173	0.778	4.960	0.000 <0.001	0.778	0.606

Tabla 6.18: UC2 - Regresión Simple entre la Utilidad Percibida y la Intención de Uso.

El modelo de regresión tiene una significancia muy alta, con  $p < 0.001$ . El  $R^2$  muestra que la variable PEOU permite explicar el 60.6 % de la varianza de ITU, lo cual indica que existe un alto grado de que ITU esté determinada por las percepciones de facilidad de uso (PEOU). Estos resultados nos permiten rechazar la hipótesis nula  $H_{70}$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la intención de uso (ITU) sí está determinada por la facilidad de uso percibida (PEOU) de los participantes.

### Intención de Uso vs Utilidad Percibida

La hipótesis H8 comprueba si las percepciones de la Intención de Uso (ITU) están determinadas por la Utilidad Percibida (PU). Para el análisis, se ha realizado la construcción de un modelo de regresión simple en donde PU es la variable independiente e ITU la variable dependiente. La ecuación de regresión resultante es la siguiente:

$$ITU = 0,591 + (0,914) * PU(11) \quad (6.12)$$

Reg. Elemento	Coef (b)	Std. E	Std. Coef	t	Sig(p)	R	R2
Constante	0.591	0.973		0.608	0.552		
PU	0.914	0.237	0.693	3.848	0.001 <0.01	0.693	0.481

Tabla 6.19: UC2 - Regresión Simple entre la Facilidad de Uso Percibida y la Intención de Uso.

El modelo de regresión es altamente significativo, con  $p < 0.01$ . El  $R^2$  muestra que la variable PU permite explicar el 48.1 % de la varianza de ITU, lo cual indica que existe un alto grado de que ITU esté determinada por las percepciones de utilidad (PU). Estos resultados nos permiten rechazar la hipótesis nula  $H_{80}$  y aceptar su hipótesis alternativa, lo que significa que se ha encontrado que la intención de uso (ITU) sí está determinada por la utilidad percibida (PU) de los participantes.



#### 6.4.4. Análisis de los resultados

En esta sección se presenta un resumen de los resultados obtenidos tras la realización de los experimentos con el objetivo de encontrar posibles semejanzas y diferencias. La Tabla 6.20 muestra cada una de las variables con los valores obtenidos, su media y desviación estándar.

Variable	UC1: Egresados y Profesionales		UC2: Estudiantes	
	Media	Desv. Est.	Media	Desv. Est.
Efectividad	0.8625	0.09498	0.7094	0.14194
Eficiencia	12.1458	2.12846	17.405	6.44788
Facilidad de uso percibida (PEOU)	4.35	0.75859	3.9556	0.71144
Utilidad percibida (PU)	4.1667	0.73168	4.0556	0.59409
Intención de uso (ITU)	4.50	0.59459	4.2963	0.7829

Tabla 6.20: Tabla de resumen de estadísticos descriptivos.

Los resultados globales nos han permitido concluir que la arquitectura de software AALIIoT ha mejorado el rendimiento de los participantes en casi la totalidad de las estadísticas analizadas.

#### Eficiencia

La Figura 6.8 presenta los diagramas de caja para la variable Eficiencia para cada cuasi-experimento. De estos diagramas se puede concluir que, bajo las condiciones de los experimentos, el diseño de sistemas usando los módulos y componentes de la arquitectura de software AALIIoT va desde los 7 hasta los 35 minutos aproximadamente.



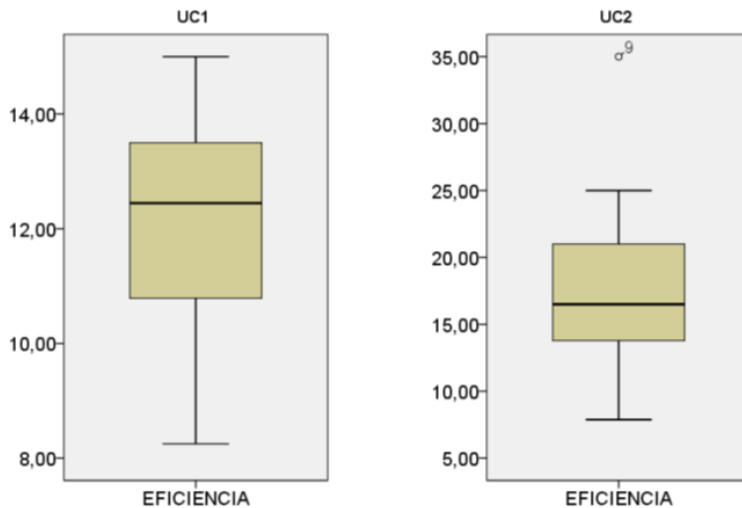


Figura 6.8: Diagrama de cajas para la variable Eficiencia en los cuasi-experimentos realizados. Fuente: Elaboración propia.

## Efectividad

La Figura 6.9 presenta los diagramas de caja para la variable Efectividad para cada cuasi-experimento. De estos diagramas se puede concluir que los participantes han sido capaces de responder efectivamente a las tareas propuesta, ejecutando la mayoría de estas de manera correcta. En general, los participantes tuvieron alrededor de un 80 % de efectividad al realizar los ejercicios experimentales.

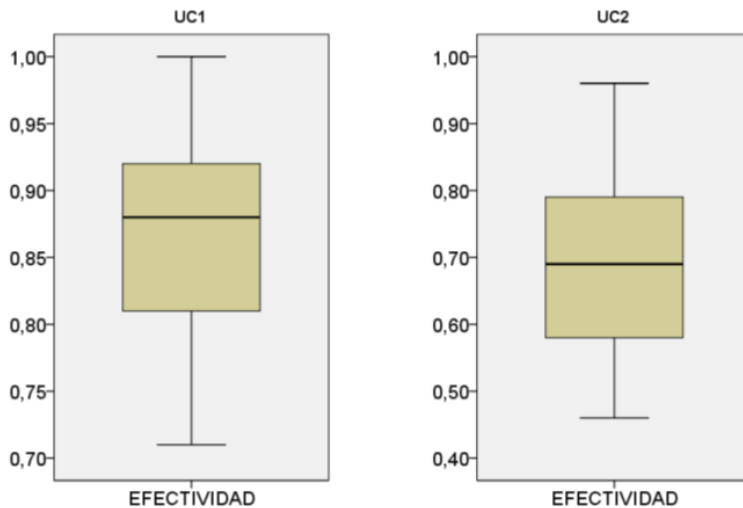


Figura 6.9: Diagrama de cajas para la variable Efectividad en los cuasi-experimentos realizados. Fuente: Elaboración propia.

### Facilidad de uso percibida (PEOU)

La Tabla 6.21 presenta un resumen de las medias obtenidas para la facilidad de uso percibida en cada cuasi-experimento. La arquitectura de software AALIIoT, en general, fue percibida como fácil de usar en relación con el valor de prueba  $v=3$ .

Variable	UC1:Egresados y Profesionales	UC2: Estudiantes
Facilidad de Uso Percibida (PEOU)	4.35	3.9556

Tabla 6.21: Tabla de resumen de medias de la Facilidad de Uso Percibida (PEOU).

### Utilidad percibida (PU)

La Tabla 6.22 presenta un resumen de las medias obtenidas para la utilidad percibida en cada cuasi-experimento. La arquitectura de software AALIIoT, en general, fue percibida como útil en relación con el valor de prueba  $v=3$ .



Variable	UC1:Egresados y Profesionales	UC2: Estudiantes
Utilidad Percibida (PU)	4.1667	4.0556

Tabla 6.22: Tabla de resumen de medias de la Utilidad Percibida (PU).

### Intención de uso (ITU)

La Tabla 6.23 presenta un resumen de las medias obtenidas para la utilidad percibida en cada cuasi-experimento. En general, los usuarios tienen la intención de utilizar la arquitectura de software AALIIoT cuando necesiten diseñar sistemas o aplicaciones enfocadas en HCI para IoT dentro de AAL en relación con el valor de prueba  $v=3$ , esto se confirma por las respuestas favorables recibidas en el cuestionario de percepción post-experimento.

Variable	UC1:Egresados y Profesionales	UC2: Estudiantes
Intención de Uso (ITU)	4.50	4.2963

Tabla 6.23: Tabla resumen de medias de la Intención de Uso (ITU).

### Resumen de aceptación de las hipótesis

La Tabla 6.24 presenta un resumen de los resultados obtenidos luego de realizar los ejercicios experimentales, para cada cuasi-experimento se indica el tipo y número de participantes y las hipótesis que fueron rechazadas.

Experi- mento	Tipo de participantes	Número de parti- cipantes	Hipótesis aceptadas	Hipótesis nulas rechazadas
UC1	Egresados y profesionales de la carrera de Ingeniería de Sistemas	12	$H_{11}, H_{21}, H_{31},$ $H_{41}, H_{50}, H_{61},$ $H_{71}, H_{81}$	$H_{10}, H_{20}, H_{30},$ $H_{40}, H_{60}, H_{70},$ $H_{80}$
UC2	Estudiantes de último año de la carrera de Ingeniería de Sistemas	18	$H_{11}, H_{21}, H_{31},$ $H_{40}, H_{50}, H_{61},$ $H_{71}, H_{81}$	$H_{10}, H_{20}, H_{30},$ $H_{60}, H_{70}, H_{80}$

Tabla 6.24: Tabla resumen de aceptación de hipótesis de los cuasi-experimentos.

Considerando los resultados de los cuasi-experimentos, se puede concluir que, en todos los casos, la arquitectura de software es percibida como fácil de usar, útil y se tiene la intención de usarla en el futuro en caso de tener la necesidad de diseñar sistemas o aplicaciones enfocadas en HCI para IoT dentro de AAL.

Respecto a la efectividad, entre los dos cuasi-experimentos, se consiguió una media de alrededor del 80 %, lo cual significa que el diseño de sistemas mediante la selección de módulos y componentes de la arquitectura de software AALIIoT se lleva a cabo de una manera correcta. Por otro lado, la eficiencia de los participantes, en los dos cuasi-experimentos, es bastante aceptable (7 a 35 minutos). Sin embargo, los resultados de la efectividad y la eficiencia podrían ser aún más favorables a medida que aumente la experiencia y entendimiento de los usuarios en la utilización de los diferentes componentes y módulos de la arquitectura, además del dominio en la que esta funciona (HCI, IoT, AAL).

## Discusión

A continuación, se presentan las conclusiones globales obtenidas de cada pregunta de investigación:

**RQ1:** *¿La arquitectura de software AALIIoT es percibida como fácil de usar y útil? De ser así, ¿las percepciones de los usuarios son el resultado de su rendimiento cuando utilizan la arquitectura para diseñar sistemas o aplicaciones enfocadas en la Interacción Hombre-Computador para aplicaciones de Internet de las Cosas en Ambientes de Vida Asistidos?*



Para responder a la primera parte de esta pregunta de investigación, es necesario referirse a las hipótesis rechazadas en los dos cuasi-experimentos realizados. En ese sentido, tanto en los cuasi-experimentos UC1 y UC2 se rechazaron las hipótesis nulas H1 y H2 aceptando sus alternativas, esto significa que, para la mayoría de participantes, la arquitectura de software fue percibida como útil y fácil de usar. Esto se ve reforzado por la efectividad alcanzada por los participantes al desarrollar los ejercicios experimentales, la cual tuvo una media del 80

Respecto a si las percepciones del usuario son el resultado de su rendimiento al utilizar la arquitectura, se puede observar que en el cuasi-experimento UC1 realizado con profesionales y egresados, la hipótesis nula H4 fue rechazada y se aceptó su alternativa, esto significa que la facilidad de uso percibida (PEOU) sí es el resultado de la eficiencia de una gran cantidad de participantes, es decir, el tiempo que los participantes emplearon en resolver los ejercicios experimentales consiguen que la arquitectura sea vista como fácil de usar. Por otro lado, en el cuasi-experimento UC2 realizado con estudiantes, se aceptó la hipótesis nula H4, lo cual significa que la facilidad de uso percibida (PEOU) no está influenciada por la eficiencia de los participantes.

Respecto a la comprobación de si la utilidad percibida es resultado de la efectividad de los participantes, en los dos cuasi-experimentos se aceptó la hipótesis nula H5; esto significa que, de manera general, la utilidad percibida (PU) no está influenciada por la efectividad de los participantes al desarrollar los ejercicios experimentales.

Por último, en los dos cuasi-experimentos, se rechazaron las hipótesis nulas H6 y se aceptaron sus alternativas, esto significa que, de forma general, la utilidad percibida (PU) sí está determinada por la facilidad de uso percibida (PEOU), es decir, los participantes consideran que, el percibir a la arquitectura como fácil de usar implica que también es útil.

**RQ2:** *¿Existe una intención de uso de la arquitectura de software AALIIoT en el futuro? De ser así, ¿tales intenciones de uso son el resultado de las percepciones de los participantes?*

Respecto a la intención de uso futura de la arquitectura propuesta, se puede observar que, para los dos cuasi-experimentos, se rechazaron las hipótesis nulas H3 y se aceptaron sus alternativas, esto significa que los participantes sí tienen la intención de usar la arquitectura en el futuro cuando exista la necesidad de diseñar y construir sistemas o aplicaciones enfocadas en HCI para IoT dentro de AAL.

Respecto a la incidencia de la facilidad de uso (PEOU) sobre la intención de uso (ITU), se observa que la hipótesis nula H7 fue rechazada y se aceptó su alternativa, esto significa que la intención de uso (ITU) sí está determinada por la facilidad de uso percibida (PEOU), es decir, los participantes consideran



que el hecho de tener una arquitectura fácil de usar implica que esta pueda ser considerada para su uso en el futuro.

Finalmente, se puede observar que la hipótesis nula H8 fue rechazada y se aceptó su alternativa, esto significa que la intención de uso (ITU) también está determinada por la utilidad percibida (PU), es decir, los participantes consideran que, si la arquitectura resulta ser útil, éstos tendrían la intención de usarla en el futuro cuando sea necesario.

Los resultados globales del análisis de regresión son resumidos en las Figuras 6.10-6.11 Estos resultados constituyen una primera aproximación empírica de la evaluación de la arquitectura propuesta. Como trabajo futuro, se presenta la necesidad de investigar sobre la influencia de otras variables basadas en el rendimiento y la percepción para predecir la aceptación de la arquitectura de software AALIIoT en la práctica.

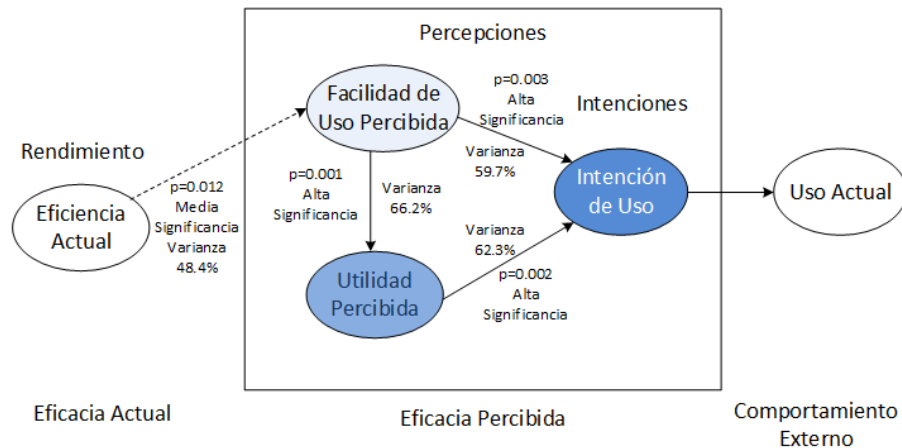


Figura 6.10: UC1 - Conclusiones de la aplicación de MEM a la Arquitectura propuesta. Fuente: Elaboración propia.

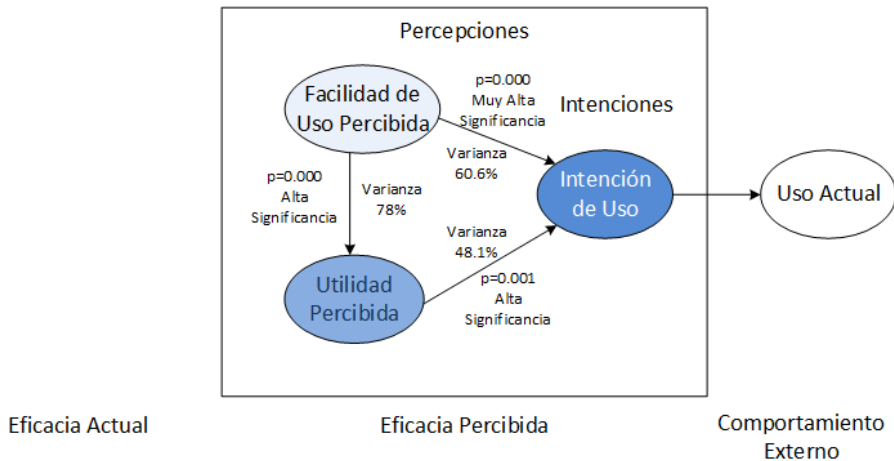


Figura 6.11: UC2 - Conclusiones de la aplicación de MEM a la Arquitectura propuesta. Fuente: Elaboración propia.

## 6.5. Amenazas a la validez

A continuación, se explican los principales problemas que pueden poner en peligro la validez del cuasi-experimento, al considerar los cuatro tipos de amenazas que se proponen en [23].

### 6.5.1. Validez interna

Las principales amenazas a la validez interna fueron: la experiencia de los participantes, los sesgos del autor y los sesgos relacionados a la forma de estructurar un sistema basado en la arquitectura propuesta usando sus componentes. Este tipo de amenaza a la validez es relevante en los estudios que tratan de establecer relaciones causales.

Para reducir la amenaza relacionada con la experiencia de los participantes, se planteó y desarrolló un ejemplo de entrenamiento representativo, el cual muestra cada paso del proceso y provee a los usuarios un alto entendimiento de la forma en la que se estructura un sistema basado en la arquitectura propuesta usando sus componentes.

Los sesgos del autor y los sesgos producidos por la entendibilidad del material fueron reducidos mediante una revisión previa de dicho material llevada a cabo por un investigador experto en el dominio de Calidad de Software e



Ingeniería de Software Empírica para así reducir posibles errores o malentendidos relacionados con el experimento, y un experimento piloto aplicado a dos sujetos de prueba con la finalidad de recibir retroalimentación que ayude a cubrir agujeros en la manera en la que se desarrollarán los experimentos.

### 6.5.2. Validez externa

Esta se refiere a la habilidad para generalizar los resultados en diferentes contextos. Según [19], la principal amenaza a la validez externa, es la representatividad de los resultados que puede verse afectada por el diseño de la evaluación, el contexto de participantes seleccionados y el tamaño y complejidad de las tareas experimentales.

El diseño de la evaluación puede tener un impacto en la generalización de los resultados debido a la complejidad de la arquitectura, sus características, módulos y componentes. Se redujo este problema proponiendo escenarios bastante descriptivos que ayuden a los sujetos a tener un panorama bastante claro de lo que se desea construir. Además, en la fase de entrenamiento se desarrollaron ejemplos en los que los evaluadores se centraron en el desarrollo de cada una de las tareas a realizar, explicándolas con un gran nivel de detalle y respondiendo dudas y cuestionamientos.

Con respecto a la experiencia de los participantes, el cuasi-experimento fue conducido con alumnos de Ingeniería de Sistemas quienes han tomado cursos de Calidad de Software e Ingeniería de software y tienen un buen conocimiento de modelos de calidad, métricas y arquitecturas de software. Además, los participantes también fueron profesionales del área de Ingeniería de Sistemas y egresados de la misma rama que, actualmente, desarrollan sus labores en análisis, diseño y programación de sistemas; además, fueron entrenados en el uso de nuestro enfoque y herramienta durante una cantidad razonable de tiempo. El tamaño y complejidad de las tareas podrían también afectar la validez externa. Para ello, hemos propuesto un conjunto de tareas experimentales con un nivel suficiente de complejidad, dado el tiempo que se tenía para las sesiones.

### 6.5.3. Validez del constructo

Las variables subjetivas están basadas en el Method Evaluation Model (MEM) [56], el cual, como se ha indicado, es un método muy bien conocido y validado empíricamente para la evaluación de tecnologías de la información. La principal amenaza es la confiabilidad del cuestionario [19]. Se ha realizado una prueba de confiabilidad del alfa de Cronbach para cada conjunto de preguntas relacionadas a cada variable subjetiva. El umbral mínimo aceptado en el campo tecnológico es  $= 0.70$ , en esta validación empírica, para la medición





de la facilidad de uso percibida (PEOU) se obtuvo un  $\alpha=0.747$ , para la medición de la utilidad percibida (PU) se obtuvo un  $\alpha=0.839$  y para la medición de la intención de uso (ITU) se obtuvo un  $\alpha=0.810$ .

#### 6.5.4. Validez de la conclusión

En este caso, los principales problemas de validez son: los tamaños de la muestra y las conclusiones estadísticas generadas usando los métodos estadísticos seleccionados, entre otros [19]. Los participantes seleccionados pertenecen a un grupo homogéneo de la misma área, mitigando así el riesgo de la variación y diferenciación de individuos. Por otro lado, para las conclusiones estadísticas, se aplicó la misma función de medición a cada variable dependiente.

### 6.6. Conclusiones

En este capítulo se ha presentado la planificación y ejecución de un cuasi-experimento y una réplica del mismo, con la finalidad de evaluar la eficacia percibida por un grupo de estudiantes, egresados y profesionales del área de Ingeniería de Sistemas durante la selección de módulos y componentes de la arquitectura propuesta para el diseño de sistemas o aplicaciones enfocados en HCI para IoT dentro de AAL.

El método que se usó para validar este enfoque fue el Method Evaluation Model (MEM). Este método considera dos aspectos importantes que son complementarios: rendimiento actual y probabilidad de aceptación en la práctica. Para la implementación de MEM se definieron variables basadas en el rendimiento (eficiencia y efectividad) como factores de influencia para las variables basadas en la percepción (facilidad de uso percibida, utilidad percibida e intención de uso).

Los resultados obtenidos revelan que: (i) la mayoría de los participantes han encontrado a la arquitectura de software AALIIoT como fácil de usar y útil; (ii) la mayoría de los participantes están de acuerdo en que, de ser necesario, sí usarían la arquitectura propuesta en el futuro; (iii) el rendimiento de los participantes en los cuasi-experimentos determinó sus percepciones positivas; (iv) las percepciones determinan la intención de usar la arquitectura propuesta. Además, los participantes indicaron, en las preguntas abiertas del cuestionario, que la arquitectura propuesta les parece útil debido a que brinda componentes y módulos que facilitan el desarrollo de sistemas o aplicaciones enfocados en HCI para IoT dentro de AAL.

Aunque los resultados son prometedores en cuanto a la utilidad de la arquitectura propuesta, estos son preliminares y deben ser considerados con cautela



ya que se basan en la evaluación de un conjunto reducido de atributos y, además, ha sido aplicado a usuarios sin experiencia previa en el desarrollo de sistemas o aplicaciones para los dominios de HCI, IoT y AAL. Por lo tanto, es necesario realizar experimentos que estén enfocados en otros aspectos de la arquitectura, como por ejemplo su implementación a nivel de programación y rendimiento de cada uno de los componentes involucrando a sujetos expertos, de preferencia profesionales de la industria.





## Capítulo 7

# Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones de este trabajo de titulación desde el punto de vista de los objetivos de investigación, su nivel de cumplimiento y los principales hallazgos obtenidos. Además, se presentan las contribuciones de esta investigación y las posibles líneas y oportunidades de investigación futura. La sección 7.1 presenta las conclusiones desde el punto de vista del objetivo general, los objetivos específicos de este trabajo y la hipótesis de investigación aceptada; finalmente, la sección 7.2 presenta los trabajos futuros.



## 7.1. Conclusiones

A continuación, se presenta la consecución de cada uno de los objetivos que fueron planteados y la hipótesis de investigación aceptada en este trabajo de titulación.

### 7.1.1. Objetivo general

El objetivo general de este trabajo fue diseñar una arquitectura de software para establecer diferentes tipos de interacciones hombre-computador (HCI) para sistemas o aplicaciones basadas en Internet de las Cosas (IoT) que se despliegan en ambientes de vida asistidos (AAL). Este objetivo ha sido cumplido en su totalidad ya que, a lo largo de este trabajo de titulación, se ha planteado el diseño de una arquitectura de software que considere los tipos de interacción natural para personas con discapacidad y adultos mayores. Los tipos de interacciones empleados fueron obtenidos como parte del resultado de un mapeo sistemático de la literatura. Además, se han definido los diferentes módulos y componentes que forman parte de la arquitectura de software AALIIoT, esto mediante un trabajo investigativo que consistió en la revisión de artículos científicos que se enfocan en proponer sistemas que contribuyen a mantener el bienestar de adultos mayores, discapacitados, personas con enfermedades crónicas, etc., en entornos como hospitales, hogares, geriátricos, etc., y considerar los componentes más importantes de cada uno de esos sistemas para su inclusión en el diseño de la arquitectura.

La arquitectura de software AALIIoT presenta varios beneficios, a continuación, se presentan algunos de ellos:

1. Su descomposición en módulos y componentes permite que la arquitectura sea flexible y pueda ser adaptada a cualquier tipo de sistema o aplicación que se despliegue dentro del dominio establecido en este trabajo de investigación (HCI, IoT, AAL).
2. La arquitectura no se limita a un protocolo de comunicación en específico, si no que permite que los componentes se comuniquen entre sí usando los protocolos que el equipo de analistas de sistemas considere adecuados para el entorno de despliegue, pudiendo usar uno o varios protocolos (WiFi, Bluetooth, Zigbee, etc.) en un mismo sistema o aplicación.
3. Los módulos y componentes de la arquitectura están pensados de tal manera que permitan su implementación usando cualquier enfoque de programación, por ejemplo, microservicios, SOA, Cliente-Servidor mediante sockets, etc.



4. Los sistemas que se desarrollen considerando esta arquitectura pueden implementar únicamente los módulos y componentes que se consideren necesarios para satisfacer los requerimientos de usuario, sin embargo, si en el futuro se desean agregar los módulos que no fueron considerados inicialmente, esta sería una tarea que no necesita de una reconstrucción total del sistema.

### 7.1.2. Objetivos específicos

Para lograr el objetivo general, se plantearon algunos objetivos específicos los cuales son analizados a continuación.

#### 7.1.2.1. Objetivo específico 1

*Analizar el estado actual de la investigación sobre las técnicas y estrategias que se utilizan para el desarrollo e implementación de sistemas basados en AAL, HCI e IoT.*

Este objetivo específico fue cumplido en su totalidad, para ello se realizó un mapeo sistemático de la literatura enfocado en encontrar los entornos donde se aplican las soluciones que consideran las técnicas y estrategias usadas para el desarrollo e implementación de sistemas basados en AAL, HCI, e IoT dentro del campo de la salud (eHealth). A qué usuario están dirigidas, cómo se lleva a cabo la recolección de información fisiológica y del entorno de los usuarios, las enfermedades, tratamientos y tipo de asistencia más relevantes usando estas soluciones, las formas de interacción disponibles para el control de sistemas basados en AAL e IoT y la manera en la que se ha estado llevando a cabo la investigación de estos temas.

Como resultados, se encontró que existe un gran interés en el campo de la investigación por crear soluciones que fomenten el cuidado de la salud de, principalmente, adultos mayores y personas con discapacidad. El monitoreo de signos vitales, control de enfermedades, control y monitoreo del entorno y la inteligencia ambiental, sea dentro del hogar, hospitales o geriátricos, son puntos abordados en los diferentes estudios primarios. Finalmente, entre los tipos de interacción que permiten las soluciones identificadas, podemos encontrar: la interacción gestual, interacción mental, interacción visual, interacción vocal, interacción basada en sensores y la interacción con interfaces gráficas. Sin embargo, son muy pocos los estudios primarios que se enfocan en brindar soluciones que permitan a las personas interactuar de forma más natural con los sistemas propuestos para que sean más accesibles a personas que no poseen todas las características necesarias para interactuar con sistemas que ofrecen, únicamente, interfaces gráficas.



### 7.1.2.2. Objetivo específico 2

*Diseñar una arquitectura que integre los módulos y componentes necesarios para permitir la interconexión de tecnologías de forma adecuada en un mismo sistema considerando el entorno o escenario de uso al que será destinada.*

Este objetivo específico fue cumplido totalmente, para ello, mediante una revisión de la literatura en la que se consideraron artículos científicos que proponen sistemas para mantener el bienestar de sectores vulnerables, se identificaron sus componentes más relevantes y que se apegaban de mejor forma a los objetivos de la arquitectura que se deseaba construir. Sobre los componentes identificados, se realizó un análisis y se definieron los cambios a los que necesitaban ser sometidos para conseguir que dichos componentes puedan ser integrados en la arquitectura. Además, se definieron componentes adicionales que se considera imprescindibles para permitir el control de interacciones que estarán disponibles a los usuarios. Con los componentes definidos, se llevó a cabo una modularización de la arquitectura, de manera que, cada módulo se encargaría de una función específica mediante los componentes adecuados.

De esta manera se diseñó una arquitectura que posee cuatro módulos fundamentales, a continuación, se describe cada uno de ellos:

1. Módulo de identificación de usuarios: se encarga de autenticar o identificar a los usuarios presentes en el entorno.
2. Módulo receptor de interacción: se encarga de captar las señales realizadas por el usuario, según la interacción permitida, mediante dispositivos receptores de interacción o sensores para su posterior ejecución.
3. Middleware de interacción: se encarga de procesar las señales recibidas, convertirlas en comandos y definir el dispositivo o dispositivos objetivo a los cuales van dirigidos.
4. Módulo Controlador de ejecución: se encarga de ejecutar los comandos enviados por el usuario sobre los dispositivos o actuadores objetivo.

### 7.1.2.3. Objetivo específico 3

*Realizar la implementación de un prototipo de un sistema que haga uso de los componentes de la arquitectura de software AALIIoT.*

Este objetivo fue cumplido totalmente, ya que se llevó a cabo el diseño, planificación e implementación del prototipo de un sistema enfocado en la Interacción Hombre-Computadora (HCI) para Internet de las Cosas (IoT) dentro de Ambientes de Vida Asistidos (AAL) que usa los módulos y componentes de la arquitectura de software AALIIoT propuesta en este trabajo de titulación.



El prototipo del sistema implementado responde a los requerimientos de un usuario dentro de un escenario específico. En este caso, el sistema satisface los requerimientos de un adulto mayor con discapacidad física, permitiéndole encender y apagar una bombilla de luz usando dos tipos de interacción:

1. gestual, usando una cámara de video y un algoritmo que identifica el gesto realizado
2. GUI, usando una aplicación móvil desplegada en un smartphone con sistema operativo Android.

Los componentes y módulos del sistema fueron implementados usando diferentes herramientas y lenguajes de programación, por ejemplo, Java, Python, Arduino, App Inventor. La comunicación entre componentes se llevó a cabo usando el paradigma Cliente-Servidor. El envío y recuperación de datos se implementó mediante solicitudes y respuestas HTTP a través de los métodos GET y POST que invocan URIs definidas en los módulos que actúan como servidores. El prototipo del sistema fue desplegado en un ambiente de pruebas con usuario reales. Los usuarios manifestaron sentirse muy conformes con las facilidades que se les brinda al poder controlar los dispositivos usando diferentes tipos de interacción ya que de esta manera se contribuye a mejorar el bienestar de las personas con discapacidades físicas y adultos mayores.

#### 7.1.2.4. Objetivo específico 4

*Evaluar la arquitectura propuesta, mediante un conjunto de escenarios, utilizando métodos de Ingeniería de Software, que permitan medir la inclusión de interacciones naturales en sistemas para ambientes de vida asistidos, por parte de personas afines al dominio de la Ingeniería de Sistemas.*

Este objetivo específico fue cumplido en su totalidad, para conseguirlo se llevó a cabo la ejecución de un cuasi-experimento y una réplica del mismo que permitieron evaluar la eficacia percibida por un grupo de estudiantes, egresados y profesionales del área de Ingeniería de Sistemas durante las tareas de selección de módulos y componentes de la arquitectura para diseñar sistemas enfocados en HCI para IoT dentro de AAL en escenarios específicos. Para realizar la validación, se usó el Method Evaluation Model (MEM) que considera dos aspectos importantes y complementarios: rendimiento actual y probabilidad de aceptación en la práctica. Para implementar del método, se definieron variables basadas en el rendimiento (eficiencia y efectividad) como factores de influencia para las variables basadas en la percepción (facilidad de uso percibida, utilidad percibida e intención de uso).

Los resultados obtenidos revelan que: (i) la mayoría de los participantes han encontrado a la arquitectura de software AALIIoT como fácil de usar y útil;





(ii) la mayoría de los participantes están de acuerdo en que, de ser necesario, sí usarían la arquitectura propuesta en el futuro; (iii) el rendimiento de los participantes en los cuasi-experimentos determinó sus percepciones positivas, es decir, la percepción de facilidad de uso de la arquitectura está determinada por la eficiencia de los participantes al desarrollar los ejercicios experimentales, mientras que, la percepción de utilidad de la arquitectura está determinada, en menor medida, por la efectividad de los participantes al desarrollar los ejercicios experimentales; (iv) las percepciones de facilidad de uso y utilidad de la arquitectura sí determinan la intención de usarla en el futuro.

### 7.1.3. Hipótesis

La elección de la hipótesis a la cual responde este proyecto de titulación, se fundamenta en los resultados obtenidos de la investigación realizada, la elaboración de la solución propuesta, la implementación de un sistema basado en dicha solución y las evaluaciones empíricas ejecutadas.

Del mapeo sistemático realizado en el Capítulo 3, se obtiene que, tanto en la industria como en la academia, existe un gran interés por crear soluciones que contribuyan al cuidado de la salud de adultos mayores y personas con discapacidad y que facilitan la realización de sus actividades cotidianas. Sin embargo, tales soluciones, no consideran todas las necesidades de interacción de las personas a las que van destinadas. En este trabajo de titulación, se propone la arquitectura de software AALIIoT, descrita en el capítulo 4, la cual contempla diferentes modos de interacción naturales y las interconecta con dispositivos que permiten a las personas realizar acciones cotidianas que, en otras circunstancias, pueden resultar difíciles de realizar.

Además, en el Capítulo 5 se presentó la implementación del prototipo de un sistema basado en la arquitectura propuesta. El prototipo funciona bajo un escenario en particular para entornos de vida asistida. Durante el despliegue, los usuarios manifestaron sentirse conformes con las funcionalidades que brinda el sistema. Por otro lado, como resultados de la evaluación empírica realizada en el Capítulo 6, se obtuvo que los participantes consideran a la arquitectura de software AALIIoT como fácil de usar, útil y que tienen la intención de usarla en el futuro cuando sea necesario, además, todas las percepciones (facilidad de uso, utilidad e intención de uso) fueron determinadas por su rendimiento positivo (efectividad y eficiencia) durante la ejecución de la evaluación.

De esta manera, se han presentado argumentos positivos y bases sólidas que permiten rechazar la hipótesis nula planteada en este trabajo de titulación y aceptar su hipótesis alternativa:

*H<sub>1</sub>: La arquitectura de software propuesta, sí brinda soporte en la creación de interacciones naturales de usuario en soluciones de Internet de las Cosas*



*para ambientes de vida asistidos.*

## 7.2. Trabajos futuros

A continuación, se plantean actividades que se podrían realizar para mejorar la arquitectura propuesta en cuanto a la arquitectura de software, instancia de la arquitectura de software y la validación empírica de la arquitectura de software.

### 7.2.1. Respecto a la arquitectura de software AALIIoT

- Crear un DSL (Lenguaje de Dominio Específico) para disminuir la brecha entre la descripción del problema y la implementación del mismo.
- Agregar un módulo que permita la comunicación con sistemas externos como, por ejemplo, un sistema hospitalario que supervisa a un paciente en el hogar.
- Realizar un mapeo de los tipos de interacción que necesitan personas con ciertas discapacidades o enfermedades. De esta forma, se podrá realizar una recomendación de la interacción adecuada para un usuario específico según la limitación que posea.

### 7.2.2. Respecto al prototipo del sistema implementado

- Incluir otros tipos de interacción de forma que el mismo sistema interactúe con varios tipos de usuarios.
- Agregar diferentes tipos de protocolos de comunicación entre los módulos.
- Optimizar los algoritmos que se pueden usar para el reconocimiento de comandos gestuales.
- Crear algoritmos que permitan el reconocimiento de comandos visuales, reconocimiento de comandos vocales y reconocimiento de comandos mentales.
- Incluir más dispositivos objetivo dentro del entorno para que el usuario los controle usando algún tipo de interacción.
- Preparar las herramientas y materiales requeridos para el despliegue y uso posterior del sistema en entornos reales.



### **7.2.3. Respecto a la validación y mejora en las evaluaciones empíricas**

- Realizar validaciones realistas aplicadas en entornos industriales reales para llevar a cabo casos de estudio o experimentos controlados.
- Desde otra perspectiva, se podría evaluar la usabilidad/experiencia de usuario en cuanto al uso de los sistemas creados haciendo uso de la arquitectura propuesta.
- Realizar evaluaciones empíricas que se centre en la comparación de la arquitectura propuesta con arquitecturas similares.



## Apéndice A

# Estudios primarios seleccionados

### A.1. IEEE Xplore

TÍTULO	AUTORES	AÑO	NRO PAGES	NRO CITAS
A Context-Aware, Interactive M-Health System for Diabetics	Shih-Hao Chang, Rui-Dong Chiang, Shih-Jung Wu, Wei-Ting Chang	2016	9	16
A Home Mobile Healthcare System for Wheelchair Users	Lin Yang, Yanhong Ge, Wenfeng Li, Wenbi Rao, Weiming Shen	2014	6	32
A Lower Limb Fracture Postoperative-guided Interactive Rehabilitation Training System and Its Effectiveness Analysis	Shih-Ching Yeh, Shun-Min Chang, Shu-Ya Chen, Wu-Yuin Hwang, Tzu-Chuan Huang, Te-Lu Tsai	2012	6	11



A Mobile Health Application for Falls Detection and Biofeedback Monitoring	Edgar T. Horta, Ivo C. Lopes, Joel J. P. C. Rodrigues, and Mario L. Proença Jr.	2013	5	12
A Mobile Healthcare Solution for Ambient Assisted Living Environments	Daniel F. M. Rodrigues, Edgar T. Horta, Bruno M. C. Silva, Fábio D. M. Guedes, and Joel J. P. C. Rodrigues	2014	6	13
A monitoring system for type 2 diabetes mellitus	Ning Wang, Guixia Kang	2012	6	8
A Motion-Sensing Enabled Personalized Exercise System for Cardiac Rehabilitation	Tung-Hung Lu, Hsing-Chen Lin, Yueh-Hsuan Lee, Rong-Rong Chen, Hsueh-Lin Chen, Shu-Yuan	2012	5	4
A Multi-communication-Fusion Based Mobile Monitoring System for Maternal and Fetal Information	Pei Lyu, Manman Peng, Yongqiang Lyu, Yu Chen, Jijiang Yang	2013	5	2
A Multi-Sensor Surveillance System for Elderly Care	Bassant Selim, Youssef Iraqi, Ho-Jin Choi	2013	5	4
A Novel System Architecture for Brain Controlled IoT Enabled Environments	Jagadish. B, M. P. R. S. Kiran, P. Rajalakshmi	2017	5	1
A Patient-Centered Medical Environment with Wearable Sensors and Cloud Monitoring	Yu-Jung Ko, Hui-Ming Huang, Wei-Han Hsing, Jerry Chou, Hung-Chih Chiu, Hsi-Pin Ma	2015	6	3



A Pharmaceutical Intelligent Information System to Detect Allergies and Adverse Drugs Reactions based on Internet of Things	Antonio J. Jara, Francisco J. Belchi, Alberto F. Alcolea, Jose Santa, Miguel A. Zamora-Izquierdo, Antonio F. G6mez-Skarmeta	2010	5	39
A Sudden Infant Death Prevention System for Babies	Ángelo M. Fonseca, Edgar T. Horta, Sandra Sendra, Joel J. P. C. Rodrigues, J. A. F. Moutinho	2014	6	6
A System for Assessment of Limb Movements in Sleep	Adriana M. Adami, André G. Adami, Tamara L. Hayes, Zachary T. Beattie	2013	5	2
A Triggering Mechanism for End-to-end IoT eHealth System with Connected Ambulance Vehicles	Amr Elsaadany, Amr Sedky, Noor Elkholy	2017	6	0
An Architecture to Manage Motor Disorders in Parkinson's Disease	Lucia Pepa, Marianna Capecciy, Federica Verdini, Maria Gabriella Ceravoloy, Luca Spalazzi	2015	6	1
An EMG-Based Human-Machine Interface to Control Multimedia Player	Mohamed Tahar Hammi, Osman Salem, Ahmed Mehaoua	2016	6	1
An IoT Based System for Remote Patient Monitoring	Alexandru Archip, Nicolae Botezatu, Elena Serban, Paul-Corneliu Herghelegiu, Andrei Zala	2016	6	9



Automated Patient Appointment Reminder for Cross-platform Mobile Application	Jirabhron Chaiwongsai, Phonthep Preecha, and Siwapan Intem	2016	6	0
Communication Protocol for Enabling Continuous Monitoring of Elderly People through Near Field Communications	Antonio J. Jara, Pablo Lopez, David Fernandez, Miguel A. Zamora, Benito Ubeda and Antonio F. Skarmeta	2013	24	15
Context-sensitive Access in Industrial Internet of Things (IIoT) Healthcare Applications	Fadi Al-Turjman	2018	8	1
Design Ubiquitous Architecture for Telemedicine Based on Mhealth Arduino 4G LTE	Yair Enrique Rivera Julio	2016	6	2
Development of a Real-Time Vital Data Collection System from Players during a Football Game	Shinsuke Hara, Tetsuo Tsujioka, Toui Kanda, Hajime Nakamura, Takashi Kawabata, Kenji Watanabe, Masanao Ise, Noa Arime, Hiroyuki Okuhata	2013	5	12
Development of Wearable Heart Disease Monitoring and Alerting System associated with Smartphone	Hyuma Watanabe, Masatoshi Kawarasaki, Akira Sato, Kentaro Yoshida	2012	6	18



Drugs Interaction Checker based on IoT	A. J. Jara, A. F. Alcolea, M. A. Zamora and A. F. Gómez Skarmeta, Mona Alsaedy	2010	8	39
Efficient and Privacy-preserving Voice-based Search over mHealth Data	Mohammad Hadian, Thamer Altuwaiyan, Xiaohui Liang, and Wei Li	2017	6	0
Embedded Gateway Services for Internet of Things Applications in Ubiquitous Healthcare	M F A Rasid, W M W Musa, N A A Kadir, A M Noor, F Touati, W Mehmood, L Khriji, A Al-Busaidi, A Ben Mnaouer	2014	5	13
Fall Detection on the Road	Felix Büsching, Henning Post, Matthias Gietzelty, and Lars Wolf, Felix Büsching, Henning Post, Matthias Gietzelty, Lars Wolf	2013	5	11
Falling Detection Using Multiple Doppler Sensors	Shoichiro Tomii, Tomoaki Ohtsuki	2012	6	20
HEMAN: Health Monitoring and Nour. An IoT based e-Health Care System for Remote Telemedicine	Chanchal Raj, Chaman Jain and Wasim Arif	2017	5	0
Home Telehealth by Internet of Things (IoT)	Dr. Salah S. Al-Majeed, Dr. Intisar S. Al-Mejibli, Prof. Jalal Karam	2015	5	12
Implementation of Virtual Sensors for Building a Sensor-Cloud Environment	Atrayee Gupta, Nandini Mukherjee	2016	8	13





Integration of wearable devices in a wireless sensor network for an E-health application	Pedro Castillejo, José-Fernán Martínez, Jesús Rodríguez-Molina, Alexandra Cuerva	2013	12	108
Intelligent Packaging and Intelligent Medicine Box for Medication Management towards the Internet-of-Things	Zhibo Pang, Junzhe Tian, Qiang Chen	2013	9	12
Intelligent Remote Control of Smart Home Devices using Physiological Parameters	David Katz, Lassad Ben Hafsia, Osman Salem, Ahmed Mehaoua	2015	6	0
Interconnected Personal Health Record Ecosystem Using IoT Cloud Platform and HL7 FHIR	Jaeki Hong, Peter Morris, Jonghwa Seo	2017	6	1
Internet of Thing Based HealthCare Monitoring System	Himadri Nath Saha, Shreyaasha Chaudhury, Ruptirtha Mukherjee, Debasmita Paul, Siddhartha Haldar	2017	5	0
IoT Based Patient Monitoring and Diagnostic Prediction Tool using Ensemble Classifier	Ani R, Krishna S, Anju N, Sona Aslam M, O.S Deepa	2017	6	0
IOT based Wearable Biomedical Monitoring System	Supriya Kale, Satendra Mane, Pravin Patil	2017	6	0
IoT-Based Fall Detection for Smart Home Environments	Shalom Greene, Himanshu Thapliyal, and David Carpenter	2016	6	7



IoT-based Remote Facial Expression Monitoring System with sEMG Signal	Mingzhe Jiang, Tuan Nguyen Gia, Arman Anzanpour, Amir-Mohammad Rahmani, Tomi Westerlund, Sanna Salanterä, Pasi Liljeberg, and Hannu Tenhunen	2016	6	8
IoT-enabled Physical Telerehabilitation Platform	Mauro Caporuscio, Danny Weyns, Jesper Andersson, Clara Axelsson and Goran Petersson	2017	8	1
Mobile Health Application for Early Disease Outbreak-Period Detection	Preetika Rani, Vaskar Raychoudhury, Sandeep Singh Sandha, Dhaval Patel	2014	6	3
Monitoring Elder's Living Activity Using Ambient and Body Sensor Network in Smart Home	Ya-Hung Chen, Ming-Je Tsai, Chia-Hui Chen, Chao-Lin Wu, Yi-Chong Zeng	2015	6	7
Multi-purpose Connected Electronic Nose System for Health Screening and Indoor Air Quality Monitoring	Jongwoo Choi, Jun Seok Park, SungJune Chang, Hae Ryong Lee	2017	5	0
MyHeart: An Intelligent mHealth Home Monitoring System Supporting Heart Failure Self-Care	Nagla Almosayan, Edward Lee, Ala Alluhaidan, Samir, Linda Houston-Feenstra, Mercy Kagoda, Wayne Dysinger	2014	6	0



Non-Intrusive Human Activity Monitoring in a Smart Home Environment	S. Mohsen Amiri, Mahsa T. Pourazad, Panos Nasiopoulos, Victor C.M. Leung	2013	5	10
Offline Domotic System using voice comands	Javier Errobidart, Alejandro José Uriz, Esteban González, Iván Exequiel Gelosi, Juan Alberto Etcheverry	2017	6	0
On Construction of an Intelligent Environmental Monitoring System for Healthcare	Chao-Tung Yang, Jung-Chun Liu, Chi-Jui Liao, Chia-Cheng Wu, Fang-Yie Leu	2013	8	4
Real Time Falls Prevention and Detection with Biofeedback Monitoring Solution for Mobile Environments	Edgar T. Horta, Ivo C. Lopes, Joel J. P. C. Rodrigues, and Sudip Misra	2013	7	15
Remote Follow Up of Health Through the Monitoring of Electrical Activities on the Residential Power Line – Preliminary Results of an experimentation	N. Noury, Senior Member, Kim Anh Quach, Marc Berenguer, Henri Teyssier, Marie-Jeanne Bouzid, Laurent Goldstein, Michel Giordani	2009	5	18
Remote Home-Based Ante and Post Natal Care	Martin Masek, Chang Su Lee, Chiou Peng Lam, Keng T. Tan	2009	6	3
Security Architecture for Mobile E-Health Applications in Medication Control	Fabio Goncalves, Joaquim Macedo, M. Joao Nicolau, Alexandre Santos	2013	8	20



SemanMedical: A Kind of Semantic Medical Monitoring System Model Based on the IoT Sensors	Guigang Zhang, Chao Li, Yong Zhang, Chunxiao Xing, Jijiang Yang	2012	6	20
Smart Self-Regulatory Health System	Chaitanya Kulkarni, Snehal Kenjale, Manasi Patil, Chinmay Kulkarni, S. R Hiray	2017	5	0
Software Defined Healthcare Networks	Long Hu, Meikang Qiu, Jeungeun Song, M. Shamim Hossain, and Ahmed Ghoneim	2015	9	39
System Architecture of Customized Intelligent Lighting Control and Indoor Environment Monitoring System for Persons with Mild Cognitive Impairment or Dementia	Mika Raatikainen, Robert Ciszek, Johanna Närvalainen, Juho Merilahti, Sami Siikanen, Timo Ollikainen, Ilona Hallikainen, Jukka-Pekka Skön	2016	6	1
The Alzimio App for Dementia, Autism & Alzheimer's: Using Novel Activity Recognition Algorithms and Geofencing	Jad Helmy, Ahmed Helmy	2016	6	3
Virtual and Augmented Reality Environment for Remote Training of Wheelchairs Users Social, Mobile, and Wearable Technologies Applied to Rehabilitation	E. L. M. Naves, T. F. Bastos, G. Bourhis, Y. M. L. R. Silva, V. F. Lucena Jr	2016	4	2



Web-of-Things Inspired e-Health Platform for Integrated Diabetes Care Management	Majid A. Al-Tae, Ala H. Sungoor, Suhail N. Abood, Nada Y. Philip	2013	6	15
--	---	------	---	----

Tabla A.1: Estudios primarios seleccionados de IEEE Xplore



## A.2. ACM Digital Library

TÍTULO	AUTORES	AÑO	NRO PAGS	NRO CITAS
A Next-Generation Augmented Reality Platform for Mass Casualty Incidents (MCI)	Fatih Demir, Salman Ahmad, Prasad Calyam, Duo Jiang, Rui Huang, Isa Jahnke	2017	22	1
A Testbed for Adaptive Security for IoT in eHealth	Yared Berhanu, Habtamu Abie, Mohamed Hamdi	2013	8	23
Application of ZigBee and RFID Technologies in Healthcare in Conjunction with the Internet of Things	Nawaf Alharbe, Anthony S. Atkins, Akbar Sheikh Akbari	2013	5	13
Designing a Context-Aware Assistive Infrastructure for Elderly Care	Simon Klakegg, Chu Luo, Jorge Goncalves, Niels van Berkel, Denzil Ferreira, Aku Visuri, Hanna-Leena Huttunen, Simo Hosio	2017	6	1
Drug identification and interaction checker based on IoT to minimize adverse drug reactions and improve drug compliance	Antonio J. Jara, Miguel A. Zamora, Antonio F. Skarmeta	2012	13	28
EcoMeal: A Smart Tray for Promoting Healthy Dietary Habits	Jaejeung Kim, Joonyoung Park, Uichin Lee	2016	6	3



Mining Minds: an innovative framework for personalized health and wellness support	Oresti Banos, Muhammad Bilal Amin, Wajahat Ali Khan, Taqdir Ali, Muhammad Afzal, Byeong Ho Kangy and Sungyoung Lee	2015	8	23
SmartHealth-NDNoT: Named Data Network of Things for Healthcare Services	Divya Saxena, Vaskar Raychoudhury, Nalluri SriMahathi	2015	6	9
UbiHeld - Ubiquitous Healthcare Monitoring System for Elderly and Chronic Patients	Avik Ghose, Priyanka Sinha, Chirabrata Bhaumik, Aniruddha Sinha, Amit Agrawal, Anirban Dutta Choudhury	2013	10	22

Tabla A.2: Estudios primarios seleccionados de ACM Digital Library



### A.3. Science Direct

TÍTULO	AUTORES	AÑO	NRO PAGES	NRO CITAS
A personal connected health system for the Internet of Things based on the Constrained Application Protocol	Danilo F.S. Santos, Hyggo O. Almeida, Angelo Perkusich	2015	15	25
The “Smartstone”: using smartphones as a telehealth gateway for senior citizens	P. Bellagente, C. Crema, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, E. Sisinni, A. Vezzoli	2016	6	7

Tabla A.3: Estudios primarios seleccionados de Science Direct





## A.4. Springer Link

TÍTULO	AUTORES	AÑO	NRO PAGES	NRO CITAS
A cloud-based virtual caregiver for elderly people in a cyber physical IoT system	Md. Abdur Rahman, M. Shamim Hossain	2018	14	1
A Context-Aware Interactive Health Care System Based on Ontology and Fuzzy Inference	Tzu-Chiang Chiang, Wen-Hua Liang	2015	25	5
A Health Gateway for Mobile Monitoring in Nursing Home	Yu Li, Pengfeng Liu, Qian Cai, Junwen Guo, Ziwei Zhou, Huan Yan, Meiyu Qian, Fengyuan Yu, Kun Yuan, Juan Yu	2018	15	0
A Multi-Constraint Scheme with Authorized Mechanism for the Patient Safety	Jen-Ming Hsu, Yao-Chang Yu, Ting-Wei Hou, Wei-Guang Teng, Tzu-Chiang Chiang, Ming-Chang Yang	2016	10	1
A novel and secure IoT based cloud centric architecture to perform predictive analysis of users activities in sustainable health centres	P. K. Gupta, B. T. Maharaj, Reza Malekian	2016	24	17



A preliminary study of telemedicine for patients with hepatic glycogen storage disease and their healthcare providers: from bedside to home site monitoring	Irene J. Hoogeveen, Fabian Peek, Foekje de Boer, Charlotte M. A. Lubout, Tom J. de Koning, Sebastiaan te Boekhorst, Robert-Jan Zandvoort, Rob Burghard, Francjan J. van Spronsen, Terry G. J. Derks	2018	8	0
A residential maintenance-free long-term activity monitoring system for healthcare applications	Xenofon Fafoutis, Evgeny Tsimbalo, Evangelos Mellios, Geoffrey Hilton, Robert Piechocki, Ian Craddock	2016	20	16
A Supplementary Automatic Door Device for Hybrid Support of Humans and Robots	Rui Fukui, Keita Kadowaki, Hiroshi Noguchi, Taketoshi Mori, Tomomasa Sato	2010	8	1
An Ambient Assisted Living Platform to Integrate Biometric Sensors to Detect Respiratory Failures for Patients with Serious Breathing Problems	Antonio J. Jara, Miguel A. Zamora, Antonio F. Gómez Skarmeta	2011	9	6
An Architecture to Combine Context Awareness and Body Sensor Networks for Health Care Applications	Alessia Salmeri, Carlo Alberto Licciardi, Luca Lamorte, Massimo Valla, Roberta Giannantonio, and Marco Sgroi	2009	8	8



An automated and intelligent Parkinson disease monitoring system using wearable computing and cloud technology	Ahmad Almogren	2017	8	0
An internet of things-based personal device for diabetes therapy management in ambient assisted living (AAL)	Antonio J. Jara, Miguel A. Zamora, Antonio F. G. Skarmeta	2011	10	212
An IoT healthcare service model of a vehicle using implantable devices	Yoon-Su Jeong, Seung-Soo Shin	2016	10	1
An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare	Zhe Yang, Qihao Zhou, Lei Lei, Kan Zheng, Wei Xiang	2016	11	27
An ontology-driven personalized food recommendation in IoT-based healthcare system	V. Subramaniaswamy, Gunasekaran Manogaran, R. Logesh, V. Vijayakumar, Naveen Chilamkurti, D. Malathi, N. Senthilselvan	2018	33	1
Anti-fall: A Non-intrusive and Real-Time Fall Detector Leveraging CSI from Commodity WiFi Devices	Daqing Zhang, Hao Wang, Yasha Wang, Junyi Ma	2015	13	17



AVATAR: An Open Source Architecture for Embodied Conversational Agents in Smart Environments	Marcos Santos-Pérez, Eva González-Parada, José Manuel Cano-García	2011	7	1
Behavioral Reference Model for Pervasive Healthcare Systems	Arezoo Tahmasbi, Sahar Adabi, Ali Rezaee	2016	23	1
Big Data and IoT for Chronic Patients Monitoring	Diego Gachet Páez, Fernando Aparicio, Manuel de Buenaga, Juan R. Ascanio	2014	8	15
Big Data, Internet of Things and Cloud Convergence – An Architecture for Secure E-Health Applications	George Suciu, Victor Suciu, Alexandru Martian, Razvan Craciunescu, Alexandru Vulpe, Ioana Marcu, Simona Halunga, Octavian Fratu	2015	8	47
Cardiac Monitoring of Marathon Runners Using Disruption-Tolerant Wireless Sensors	Djamel Benferhat, Frédéric Guidec, Patrice Quinton	2012	8	8
Cloud based u-healthcare network with QoS guarantee for mobile health service	Kyungyong Chung, Roy C. Park	2017	15	2
Cloud-Assisted Mood Fatigue Detection System	Xiaobo Shi, Yixue Hao, Delu Zeng, LuWang, M. Shamim Hossain, Sk Md MizanurRahman, Abdulhameed Alelaiwi	2016	9	1



Cloud-centric IoT based student healthcare monitoring framework	Prabal Verma, Sandeep K. Sood, Sheetal Kalra	2017	17	5
Context-Awareness as an Enhancement of Brain-Computer Interfaces	Agustin A. Navarro, Luigi Ceccaroni, Filip Velickovski, Sergi Torrellas, Felip Miralles, Brendan Z. Allison, Reinhold Scherer, Josef Faller	2011	8	14
Context-Based Fall Detection Using Inertial and Location Sensors	Hristijan Gjoreski, Mitja Luštrek, Matjaž Gams	2012	16	21
Design and evaluation of a ubiquitous chest-worn cardiopulmonary monitoring system for healthcare application: a pilot study	Jiewen Zheng, Congying Ha, Zhengbo Zhang	2016	12	2
Design and Validation of Android Based Wireless Integrated Device for Ubiquitous Health Monitoring	Mandeep Singh, Neelu Jain	2015	14	2
Design of Novel Feeding Robot for Korean Food	Won-Kyung Song, Jongbae Kim, Kwang-Ok An, In-Ho Lee, Won-Jin Song, Bum Suk Lee, Sung-Il Hwang, Mi-Ok Son, Eun-Chang Lee	2010	8	7



Designing an artificial pancreas architecture: the AP@home experience	Giordano Lanzola, Chiara Toffanin, Federico Di Palma, Simone Del Favero, Lalo Magni, Riccardo Bellazzi	2014	13	9
Development of a Multi-Agent m-Health Application Based on Various Protocols for Chronic Disease Self-Management	Hyun Sang Park, Hune Cho, Hwa Sun Kim	2015	14	16
Drug identification and interaction checker based on IoT to minimize adverse drug reactions and improve drug compliance	Antonio J. Jara, Miguel A. Zamora, Antonio F. Skarmeta	2012	13	28
EMon: Embodied Monitorization	Davide Carneiro, Paulo Novais, Ricardo Costa, Pedro Gomes, José Neves	2009	10	19
eWALL: An Intelligent Caring Home Environment Offering Personalized Context-Aware Applications Based on Advanced Sensing	Sofoklis Kyriazakos, Mihail Mihaylov, Bayu Anggorojati, Albena Mihovska, Razvan Craciunescu, Octavian Fratu, Ramjee Prasad	2016	19	20



eWALL: An Open-Source Cloud-Based eHealth Platform for Creating Home Caring Environments for Older Adults Living with Chronic Diseases or Frailty	Sofoklis Kyriazakos, Ramjee Prasad, Alben Mihovska, Aristodemos Pnevmatikakis, Harm op den Akker, Hermie Hermens, Paolo Barone, Alessandro Mamelli, Samuele de Domenico, Matthias Pocs, Andrej Grguric, Miran Mosmondor, Dina Simunic, Antun Kerner, Nikola Zaric, Milica Pejanovic -Djurisic, Vladimir Poulkov, Krasimir Tochev, Beatrix Zechmann, Markus Garschall, Angeliki Angeletou, Stefano Bonassi, Francesco Infarinato, Octavian Fratou, Alexandru Vulpe, Carmen Voicu, Liljana Gavrilovska, Vladimir Atanasovski	2017	41	0
Exploring Temporal Analytics in Fog-Cloud Architecture for Smart Office HealthCare	Munish Bhatia, Sandeep K. Sood	2018	19	0
Health Fog: a novel framework for health and wellness applications	Mahmood Ahmad, Muhammad Bilal Amin, Shujaat Hussain, Byeong Ho Kang, Taechoong Cheong, Sungyoung Lee	2016	19	27



Healthcare-based on Cloud Electrocardiogram System: A Medical Center Experience in Middle Taiwan	Xiu-Shu Zhang, Fang-Yie Leu, Ching-Wen Yang, Lai-Shiun Lai	2018	13	0
Improving medication adherence in home care using a bidirectional medication assistance system	Shiori Suzuki, Jin Mitsugi, Jun Murai	2013	8	0
Intelligent face recognition and navigation system using neural learning for smart security in Internet of Things	Priyan Malarvizhi Kumar, Ushadevi Gandhi, R. Varatharajan, Gunasekaran Manogaran, Jidhesh R., Thanjai Vadivel	2017	12	25
Intelligent Information System to Tracking Patients in Intensive Care Units	Fernando Marins, Luciana Cardoso, Filipe Portela, Manuel Santos, António Abelha, José Machado	2013	8	6
Internet of things for remote elderly monitoring: a study from user-centered perspective	Iman Azimi, Amir M. Rahmani, Pasi Liljeberg, Hannu Tenhunen	2016	17	15
IoT-based cloud framework to control Ebola virus outbreak	Sanjay Sareen, Sandeep K. Sood, Sunil Kumar Gupta	2016	18	4
IoT-Based Healthcare Framework for Biomedical Applications	Faruk Aktas, Celal Ceken, Yunus Emre Erdemli	2017	14	0





JeWheels: Kinect Based Serious Game Aimed at Wheelchair Users	Zelai Sáenz de Urturi, Amaia Méndez Zorrilla, Begoña Garca Zapirain	2012	8	7
M2M-based smart health service for human UI/UX using motion recognition	Roy C. Park, Hoill Jung, Dong-Kun Shin, Gui-Jung Kim, Kun-Ho Yoon	2014	12	29
Mining-based lifecare recommendation using peer-to-peer dataset and adaptive decision feedback	Hyun Yoo, Kyungyong Chung	2017	12	1
Mobile Personal Health Care System for Patients with Diabetes	Fuchao Zhou, Hen-I Yang, José M. Reyes Álamo, Johnny S. Wong, Carl K. Chang	2010	8	38
Monitoring Patient Recovery Using Wireless Physiotherapy Devices	Nirmalya Roy, Brooks Reed Kindle	2015	9	1
Multimodal Laser-Vision Approach for the Deictic Control of a Smart Wheelchair	Frédéric Leishman, Odile Horn, Guy Bourhis	2009	10	8
Multimodal Situational Awareness for Eldercare	Weimin Huang, Aung Aung Phyo Wai, Siang Fook Foo, Jit Biswas, Chi-Chun Hsia, Koujuch Liou	2010	9	3
New secure healthcare system using cloud of things	Ebrahim Al Alkeem, Dina Shehada, Chan Yeob Yeun, M. Jamal Zemerly, Jiankun Hu	2017	19	6



Personalized HealthMonitoring Systemfor ManagingWell-Being in Rural Areas	Prema Nedungadi, Akshay Jayakumar, Raghu Raman	2017	11	1
PHR Based Diabetes Index Service Model Using Life Behavior Analysis	Hyun Yoo, Kyungyong Chung	2016	14	14
Remote Monitoring Using Smartphone Based Plantar Pressure Sensors: Unimodal and Multimodal Activity Detection	Ferdaus Kawsar, Sheikh Ahamed, Richard Love	2015	9	3
Results of Mobility and Obstacles Detection of an Experimental Three-Legged Prototype for Blind and Deafblind People	J. Alberto Garcia, Javier Poncela	2011	8	0
SeSaMoNet 2.0: Improving a Navigation System for Visually Impaired People	Ugo Biader Ceipidor, Carlo Maria Medaglia, Eliseo Sciarretta	2010	6	6
Situated Agents and Humans in Social Interaction for Elderly Healthcare: From Coaalas to AVICENA	Ignasi Gómez-Sebastià, Jonathan Moreno, Sergio Álvarez-Napagao, Dario Garcia Gasulla, Cristian Barrué, Ulises Cortés	2015	20	3



Smart Clothing: Connecting Human with Clouds and Big Data for Sustainable Health Monitoring	Min Chen, Yujun Ma, Jeungeun Song, Chin-Feng Lai, Bin Hu	2016	21	93
Smart Fall: Accelerometer-Based Fall Detection in a Smart Home Environment	Dennis Sprute, Aljoscha Portner, Alexander Weinitschke, Matthias König	2015	12	6
Smart NFC-sensors for healthcare applications and further development trends	S. Cecil, M. Bammer, G. Schmid OVE, K. Lamedschwandner OVE, A. Oberleitner	2013	10	2
Smartphone-Based System for Sensorimotor Control Assessment, Monitoring, Improving and Training at Home	Quentin Mourcou, Anthony Fleury, Céline Franco, Bruno Diot, Nicolas Vuillerme	2015	11	0
Supporting Technology for Wheelchair Users Intuitive Interface and Step Climbing Assistance	Sho Yokota, Hiroshi Hashimoto, Daisuke Chugo, Yasuhiro Ohyama, Jinhua She, Hisato Kobayashi	2013	8	0
The Smart Health Initiative in China: The Case of Wuhan, Hubei Province	Meiyu Fan, Jian Sun, Bin Zhou, Min Chen	2015	17	15
Towards Robot-Assisted Echocardiographic Monitoring in Catheterization Laboratories	Christina Pahl, Henning Ebel, Mostafa Sayahkaraji, Eko Supriyanto, Amiliana Soesanto	2017	15	0



User Interaction Concepts in Smart Caring Homes for Elderly with Chronic Conditions	Cristian-Dan Bara, Miriam Cabrita, Harm op den Akker, Hermie J. Hermens	2015	12	0
---	--	------	----	---

Tabla A.4: Estudios primarios seleccionados de Springer Link.



## Apéndice B

# Porcentajes de los criterios de extracción

Cod.	Criterios	Posibles respuestas	Estudios	%
RQ1: ¿En qué entornos se aplican soluciones basadas en IoT dentro del área de la salud y a quiénes están dirigidas?				
EC1	ENTORNOS	Hospital	41	24 %
		Hogar	97	56 %
		Áreas de recreación	17	10 %
		Vehículo	4	2 %
		Laboral	1	1 %
		Ciudad	11	6 %
		Farmacias y geriátricos	2	1 %
EC2	SECTOR VULNERABLE	Adultos mayores	42	66 %
		Personas con discapacidad	18	28 %
		Otros	4	6 %
EC3	STAKEHOLDERS	Pacientes Crónicos	25	16 %
		Pacientes Ambulatorios	67	43 %
		Cuidadores	24	15 %
		Doctores	32	21 %
		Otros	8	5 %

Tabla B.1: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ1



Cod.	Criterios	Posibles respuestas	Estudios	%
RQ2: ¿Cómo se lleva a cabo la recolección de datos fisiológicos y del entorno de una persona para el uso dentro del área de la salud?				
EC4	ACCESORIOS	Accesorios de ropa	12	11 %
		Accesorios para el cuerpo	61	57 %
		Accesorios para electrodomésticos	2	2 %
		Otros accesorios para el hogar	20	19 %
		Accesorios para fuera del hogar	2	2 %
		Accesorios para hospitales	10	9 %
EC5	EQUIPOS DE RECOLECCIÓN DE DATOS	Monitor médico	33	20 %
		Sensores	95	58 %
		Smartphone	24	15 %
		Kinect	7	4 %
		Otros dispositivos de captación gestual	1	1 %
		Eyetracker	1	1 %
		Gafas de realidad virtual	1	1 %
		Smart Glasses	1	1 %
EC6	TIPOS DE SENSORES Y MONITORES	Sensor térmico	45	12 %
		Sensor acústico	12	3 %
		Sensor de movimiento	42	11 %
		Otros sensores de medio ambiente	15	4 %
		Sensor de monitoreo respiratorio	32	9 %
		Sensor magnético	4	1 %
		Sensor óptico	11	3 %
		Sensor para alcohol y cigarro	2	1 %
		Sensor de navegación	29	8 %
		Sensor de transpiración	7	2 %
		Sensor y monitor cardíaco	56	15 %
		Sensor de monitoreo hemodinámico	34	9 %
		Monitor neurológico	6	2 %
		Monitor muscular	14	4 %
		Monitor ocular	1	0 %



	Sensor de glucosa	22	6 %
	Sensor de Visión Artificial	15	4 %
	Otros	28	7 %

Tabla B.2: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ2



Cod.	Criterios	Posibles respuestas	Estudios	%
RQ3: ¿Qué aplicaciones, implementaciones, soluciones o sistemas propuestos basados en IoT existen para el área de la salud?				
EC7	SISTEMAS	Context-aware	8	4 %
		Ambiente inteligente	14	8 %
		Rehabilitación física	8	4 %
		Monitoreo de signos vitales	62	34 %
		Monitoreo y control de enfermedades	19	10 %
		Control del entorno	5	3 %
		Monitoreo de medicación	10	5 %
		Detección y prevención de caídas	7	4 %
		Monitoreo del entorno	13	7 %
		Monitoreo de actividad humana	9	5 %
		Sistema de información hospitalario	3	2 %
		Cuidado pre y postnatal	2	1 %
		Control alimenticio	3	2 %
		Detección de obstáculos	3	2 %
		Otros	19	10 %
EC8	TIPO DE APLICACIÓN DE SOFTWARE	App	73	27 %
		Web	48	18 %
		Desktop	27	10 %
		Mobile	18	7 %
		Sensores Virtuales	1	0 %
		Infraestructura cloud	44	16 %
		Base de datos	59	22 %
EC9	NIVEL DE IMPLEMENTACIÓN	Nivel de datos	105	34 %
		Nivel de algoritmos	100	32 %
		Nivel GUI	106	34 %
EC10	PROTOCOLO DE COMUNICACIÓN	GPRS	13	5 %
		IEEE 802.15.4	11	4 %
		ZigBee	24	10 %
		XBee	4	2 %
		Bluetooth	48	19 %
		RFID	17	7 %
		WiFi	69	28 %
		NFC	9	4 %





		6LoWPAN	8	3 %
		USB	21	8 %
		HL7 FHIR	4	2 %
		GSM	12	5 %
		HTTP	2	1 %
		IrDA	2	1 %
		Otro	4	2 %

Tabla B.3: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ3

Cod.	Criterios	Posibles respuestas	Estudios	%
RQ4: ¿Cuáles son las enfermedades, tratamientos y tipos de asistencia más relevantes que pueden ser cubiertos usando soluciones IoT en el área de la salud?				
EC11	ENFERMEDADES	Enfermedades endocrinas	7	27 %
		Enfermedades del sistema cardiocirculatorio	6	23 %
		Enfermedades del sistema respiratorio	2	8 %
		Enfermedades del sistema nervioso	4	15 %
		Desórdenes mentales	3	12 %
		Lesiones físicas	4	15 %
EC12	TRATAMIENTOS	Emergencia	7	5 %
		Discapacidad	8	6 %
		Monitoreo y control de enfermedades	91	69 %
		Rehabilitación física	9	7 %
		Control de medicamentos	10	8 %
		Control alimenticio	5	4 %
		Otros	1	1 %
EC13	ASISTENCIA	Asistencia remota	92	78 %
		Asistencia personal	26	22 %

Tabla B.4: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ4



Cod.	Criterios	Posibles respuestas	Estudios	%
RQ5: ¿Cuáles son las formas de interacción disponibles para el control de soluciones propuestas basadas en IoT en el área de la salud?				
EC14	INTERACCIÓN	Gestual	9	4 %
		Mental	3	1 %
		Verbal	8	4 %
		Visión	4	2 %
		Auditivo	8	4 %
		Basadas en sensores y sensibles al contexto	95	44 %
		Interfaces WIMP y Touch	87	41 %

Tabla B.5: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ5

Cod.	Criterios	Posibles respuestas	Estudios	%
RQ6: ¿De qué manera se ha llevado hasta ahora la investigación en estos temas?				
EC15	AMBIENTE DE CREACIÓN	Industria	29	22 %
		Academia	86	64 %
		Ambos	19	14 %
EC16	MODELO DE SOLUCIÓN PROPUESTA	Metodología	4	3 %
		Proceso	3	2 %
		Prototipo	119	84 %
		Estudio experimental	4	3 %
		Estudio descriptivo	11	8 %
EC17	NOVEDAD	Nuevo	126	98 %
		Extensión	3	2 %
EC18	TIPO DE VALIDACIÓN	Prueba de concepto	56	42 %
		Caso de estudio	12	9 %
		Estudio piloto	5	4 %
		Experimento controlado	56	42 %
		Quasi-Experimento	4	3 %
EC19	COMITÉ DE ÉTICA	Si	9	7 %
		No	114	93 %

Tabla B.6: Porcentajes individuales para los criterios de extracción de la sub-pregunta RQ6



## Apéndice C

# Diagramas UML

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un conjunto de normas y estándares gráficos respecto a cómo se deben representar los esquemas relativos al software. Los diagramas UML brindan un modelado visual común para describir la arquitectura, el diseño y la implementación de sistemas de software complejos, a nivel de estructura y de comportamiento.

### C.1. Diagrama de componentes

Forma parte del conjunto de diagramas definidos por UML. Muestra los elementos de diseño de un sistema o arquitectura de software permitiendo visualizar su estructura de alto nivel y el comportamiento del servicio que sus componentes proporcionan y usan a través de interfaces. La Figura C.1 muestra un ejemplo de un diagrama de componentes básico que representa los componentes que forman parte de un proceso de ventas online.

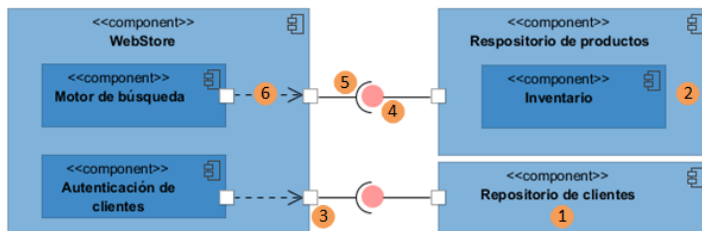


Figura C.1: Ejemplo de diagrama de componentes



La Tabla C.1 muestra los elementos más comunes usados en un diagrama de componentes, en base al diagrama ejemplo mostrado en la figura anterior.

FORMA	ELEMENTO	DESCRIPCIÓN
1	Componente	Es una parte o función reutilizable del sistema o arquitectura. Un componente proporciona y consume servicios a través de interfaces y puede usar otros componentes.
2	Subsistema	Representa a un componente que contiene otros componentes.
3	Puerto	Representa el punto por el cual los componentes exponen sus servicios.
4	Interfaz que provee	Representa un conjunto de servicios que implementa un componente y que pueden usar otros componentes o sistemas externos.
5	Interfaz requerida	Representa el llamado a los servicios brindados por otros componentes.
6	Dependencia	Se puede usar para indicar que una interfaz requerida por un componente se puede satisfacer mediante una interfaz proporcionada en otro.

Tabla C.1: Elementos utilizados en un diagrama de componente

## C.2. Diagrama de actividades

Los diagramas de actividades presentan un proceso de negocio o un proceso de software mostrando la forma en la que se construyen los diferentes flujos de trabajo o los procesos a través de una serie de acciones realizadas por personas, componentes de software o equipos. Además, muestra la manera en la que inician las acciones, los caminos alternativos que se pueden tomar desde el inicio hasta el fin y dónde puede ocurrir el procesamiento paralelo durante la ejecución. Este tipo de diagramas modelan el comportamiento de un sistema (procesos y flujos) a un alto nivel. La Figura C.2 presenta un diagrama de actividades básico referente al flujo de trabajo necesario para generar una factura.

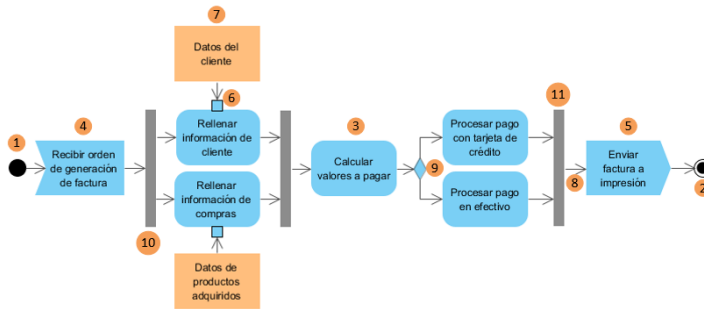


Figura C.2: Ejemplo de diagrama de actividades

La Tabla C.2 muestra los elementos más comunes usados en un diagrama de actividades, en base al diagrama ejemplo mostrado en la figura anterior.

FORMA	ELEMENTO	DESCRIPCIÓN
1	Nodo inicial	Indica el punto inicial de la actividad.
2	Nodo final	Indica el punto final de la actividad.
3	Acción	Representa a una unidad de ejecución básica.
4	Acción: aceptar evento	Unidad de ejecución que representa la aceptación de un evento o solicitud.
5	Acción: enviar señal	Unidad de ejecución que representa la señal de envío de un evento o solicitud.
6	Pin de entrada	Representa los datos que puede recibir una acción cuando se ejecuta.
7	Nodo objeto/-parámetro	Información que utilizan o generan las diferentes acciones.
8	Flujo de control	Conector que muestra el flujo de control entre las acciones.
9	Nodo de decisión	Elección de una salida entre un conjunto de posibles salidas.
10	<i>Fork Node</i>	Indica un conjunto de actividades que pueden realizarse en paralelo.
11	<i>Join Node</i>	Indica que, para continuar, deben finalizar todas las actividades concurrentes.

Tabla C.2: Elementos utilizados en un diagrama de actividades

### C.3. Diagrama de caso de uso

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En ingeniería de software un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. En otras palabras, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo. La Figura C.3 presenta un diagrama de caos de uso básico referente al requerimiento de ordenar comida.

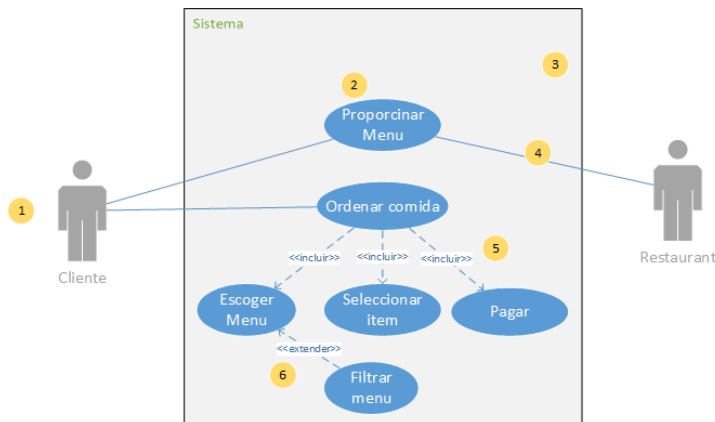


Figura C.3: Ejemplo de diagrama de caso de uso

La Tabla C.3 muestra los elementos más comunes usados en un diagrama de caso de uso, en base al diagrama ejemplo mostrado en la figura anterior.



FORMA	ELEMENTO	DESCRIPCIÓN
1	Actor	Representa un usuario, organización o sistema externo que interactúa con su aplicación o sistema
2	Caso de uso	Representa las acciones realizadas por uno o más actores en la búsqueda de un objetivo en particular
3	Sistema	El sistema o la aplicación en la que está trabajando, o parte de ella.
4	Asociación	Representa un conjunto de servicios que implementa un componente y que pueden usar otros componentes o sistemas externos.
5	Incluir	Es una forma de interacción o creación, un caso de uso dado puede incluir otro caso de uso (si el actor realiza el caso de uso base tendrá que realizar también el caso de uso incluido).
6	Extender	Es otra forma de interacción, un caso de uso dado (la extensión) puede <i>extender</i> a otro.

Tabla C.3: Elementos utilizados en un diagrama de caso de uso

## C.4. Diagrama de secuencia

Un diagrama de secuencia muestra las interacciones de los objetos dispuestas en secuencia temporal. Representa los objetos y las clases involucradas en el escenario y la secuencia de mensajes intercambiados entre los objetos necesarios para llevar a cabo la funcionalidad del escenario. Los diagramas de secuencia están típicamente asociados con realizaciones de casos de uso en la vista lógica del sistema que se está desarrollando. Un diagrama de secuencia muestra, como líneas verticales paralelas (líneas de vida), diferentes procesos u objetos que viven simultáneamente, y, como flechas horizontales, los mensajes intercambiados entre ellos, en el orden en que ocurren. Esto permite la especificación de escenarios de tiempo de ejecución simples de una manera gráfica. La Figura C.4 presenta un diagrama de secuencia básico referente a una transacción de pedido de un almacén.

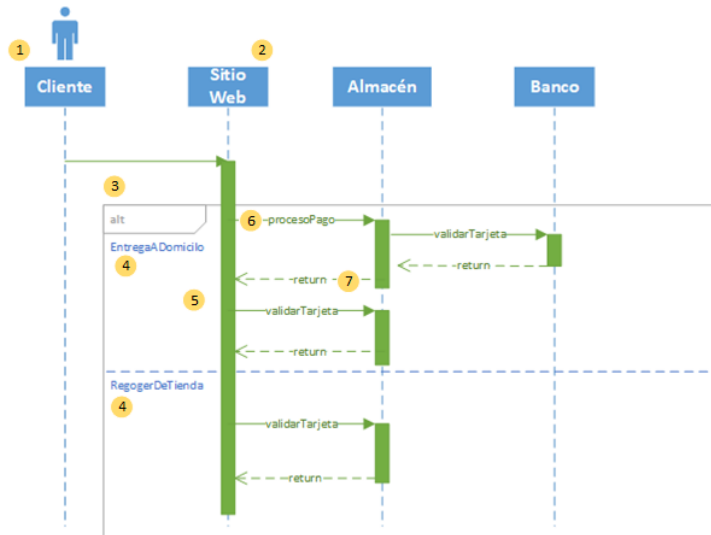


Figura C.4: Ejemplo de diagrama de secuencia

La Tabla K-4 muestra los elementos más comunes usados en un diagrama de secuencia, en base al diagrama ejemplo mostrado en la figura anterior.

FORMA	ELEMENTO	DESCRIPCIÓN
1	Actor	Participante externo al sistema que está desarrollando.
2	Línea de vida	Línea vertical que representa la secuencia de eventos que se producen en un participante durante una interacción, mientras el tiempo avanza por la línea.
3	Alternativa	Denota que se puede usar para mostrar que pueden ocurrir secuencias alternativas de mensajes.
4	Condición	Describe la condición bajo la cual se realizarán los mensajes dentro del operando de interacción.
5	Ocurrencia de ejecución	Rectángulo sombreado vertical que aparece en la línea de vida de un participante y representa el período en el que el participante ejecuta una operación.





6	Mensaje	El remitente espera una respuesta a un mensaje sincrónico antes de continuar.
7	Mensaje de devolución	Mensaje que se devuelve a un participante que espera la devolución de una llamada anterior

Tabla C.4: Elementos utilizados en un diagrama de secuencia



# Apéndice D

## Herramientas

### D.1. AppInventor

App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil. Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

### D.2. Módulo Wifi ESP8266

ESP8266 es el nombre de un microcontrolador diseñado por Espressif Systems, una compañía china. El ESP8266 se anuncia a sí mismo como una solución autónoma de redes WiFi que se ofrece como un puente entre los microcontroladores que hasta ahora existían hasta los MCU con WiFi, siendo además capaz de ejecutar aplicaciones independientes. En decir es un chip de bajo costo WIFI con una pila TCP/IP completa y un microcontrolador.

### D.2.1. Características

- Utiliza una CPU Tensilica L106 32-bit
- Voltaje de operación entre 3V y 3,6V
- Corriente de operación 80 mA
- Temperatura de operación -40°C y 125°C
- Soporta IPv4 y los protocolos TCP/UDP/HTTP/FTP
- Puertos GPIO (de propósito general) 17
- Velocidad del procesador 80 – 160 MHz

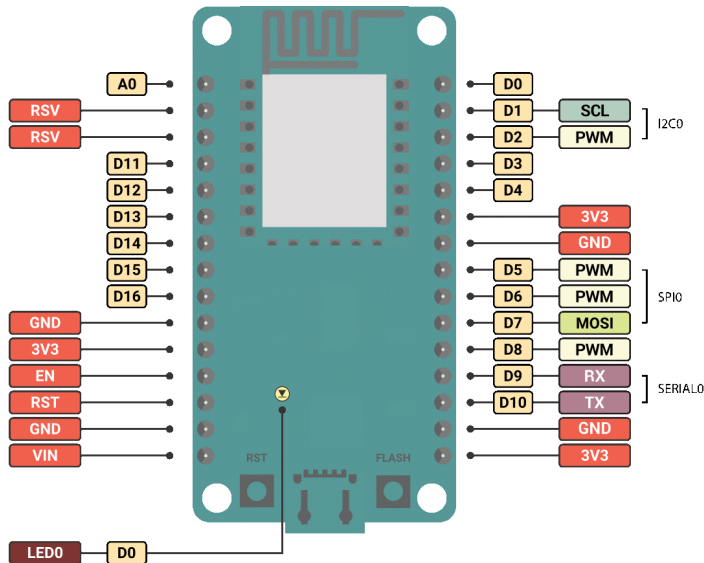


Figura D.1: Distribución de los pines del módulo WiFi

### D.3. Módulo Relé

Esta es una placa de interfaz de relé con uno o varios canales de 5V y 10A. Se puede usar para controlar varios dispositivos y otros equipos con gran corriente. Se puede controlar directamente con señales lógicas de 3.3V o 5V



desde un microcontrolador (Arduino, Raspberry Pi, ESP8266). Distribución de los pines del módulo relé:

- GND - Conecta 0V a este pin.
- IN - Controles de relé. El relé se encenderá cuando esta entrada sea inferior a 2.0V
- VCC - Conecta 5V a este pin. Se usa para alimentar los optoacopladores

## D.4. IDE Arduino

El entorno de desarrollo integrado de Arduino ( IDE ) es una aplicación multiplataforma (para Windows , macOS , Linux ) que está escrita en el lenguaje de programación Java. Una de las principales atracciones del Arduino es su baja complejidad, permitiendo a cada uno la capacidad de construir algo rápidamente y fácilmente, además que es gratuito. Se ha creado un proyecto Open Surce en Github quw proporciona una extensión(plug-in) a la herramienta al IDE, que permite escribir bocetos en el IDE de Arduino que aprovechan las interfaces de la biblioteca Arduino que, en tiempo de compilación y despliegue, generan código que se ejecutará en el ESP8266. Lo que esto significa efectivamente es que podemos usar el IDE de Arduino y construir aplicaciones ESP8266 como si fuera un Arduino.



## Apéndice E

# Experimento Empírico



## E.1. Boletín de entrenamiento

**ARQUITECTURA DE SOFTWARE AALIIoT:  
ENFOCADA EN LA INTERACCIÓN HOMBRE-  
COMPUTADOR PARA APLICACIONES DE  
INTERNET DE LAS COSAS EN AMBIENTES DE  
VIDA ASISTIDOS**

EVALUACIÓN EMPÍRICA





a cabo para dar solución a un ejercicio propuesto, en el Anexo III puede encontrar las ilustraciones que representan el conjunto de pasos que usted debe seguir de forma resumida.

#### Paso 1. Datos generales.

Escribir la información solicitada en el encabezado de la hoja de resolución (Figura 2).

<b>ESCENARIO 1</b>	
ID: _____	FECHA: _____
NOMBRE: _____	PROFESIÓN/OCUPACIÓN: _____

Figura 2. Paso uno - Datos generales en el encabezado de la hoja de resolución.

#### Paso 2. Lectura del ejercicio propuesto.

Usted debe leer el ejercicio propuesto detenidamente para su correcta comprensión (Figura 3).

**Ejercicio ejemplo**

Alex es una persona con discapacidad física: no posee sus dos piernas. Alex desea encender su radio personal y sintonizar una emisora sin necesidad de acercarse a ella. La radio se encenderá únicamente cuando sea Alex el que dé la orden haciendo caso omiso a las solicitudes de otras personas. Usar los componentes de la arquitectura que considere adecuados para construir la aplicación IoT apropiada considerando las necesidades de interacción del individuo.

Figura 3. Paso dos - Ejercicio ejemplo basado en un escenario específico.

#### Paso 3. Desarrollo de la tarea 1.

Esta tarea consiste en llenar el espacio del recuadro del Módulo Receptor de Interacción en la hoja de resolución, para ello debe realizar lo siguiente:

1. Escriba la hora en la que usted dará inicio con la tarea (Figura 4a).
2. Según el escenario propuesto, usted debe determinar cuáles son los tipos de interacción que los sujetos necesitan considerando sus restricciones físicas y/o psicológicas. Para este ejemplo, el usuario podría encender su radio usando interacciones gestuales y vocales.
3. Ahora, considerando las interacciones elegidas, usted debe determinar cuál será el dispositivo que recibirá la señal. Por ejemplo, para la interacción gestual una cámara o un Kinect; para una interacción vocal un micrófono. En este caso seleccionamos un Kinect y un micrófono (Figura 4b).
4. Ahora, en el recuadro del Módulo Receptor de Interacción, usted debe dibujar un rectángulo para cada elemento seleccionado y escribir su respectivo nombre (Figura 4c).
5. Finalmente, escribir la hora de finalización de la tarea (Figura 4d).





**ESCENARIO 1**

ID: \_\_\_\_\_ FECHA: \_\_\_\_\_  
 NOMBRE: \_\_\_\_\_ PROFESIÓN/OCUPACIÓN: \_\_\_\_\_

<p style="text-align: center;"><b>MÓDULO RECEPTOR DE INTERACCIÓN</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 80%;"> <p>Kinect </p> <p>Micrófono </p> </div> <p style="text-align: center;">a) <span style="margin-left: 100px;">d)</span></p> <p>HORA INICIO: 1:05:28 HORA FIN: 1:20:18</p>	<p style="text-align: center;"><b>MIDDLEWARE CONTROLADOR DE INTERACCIÓN</b></p> <div style="border: 1px solid black; height: 100px; margin: 10px auto; width: 80%;"></div> <p>HORA INICIO: _____ HORA FIN: _____</p>	<p style="text-align: center;"><b>MÓDULO CONTROLADOR DE EJECUCIÓN</b></p> <div style="border: 1px solid black; height: 100px; margin: 10px auto; width: 80%;"></div> <p>HORA INICIO: _____ HORA FIN: _____</p>
<p><b>MÓDULO DE IDENTIFICACIÓN DE USUARIOS</b></p> <div style="border: 1px solid black; height: 40px; margin: 10px auto; width: 80%;"></div> <p>HORA INICIO: _____ HORA FIN: _____</p>		

Identificación de usuario	Receptor de interacción	Middleware controlador de interacción	Controlador de ejecución	Dispositivo/Actuador
Rangos faciales Retina Huella Dactilar Usuario/Clave Voz	b) Termómetro Cámara Micrófono Kinect Lector de retina	Casco BCI GUI Eye Tracker Lector de huella Controlador mental Controlador gestual Controlador vocal Controlador sensores Controlador visual Controlador GUI Identificador de dispositivo objetivo	Arduino Raspberry ESP8266 REGISTRO _____ _____	Foco Puerta Ventilador Calefactor TV Radio

Figura 4. Paso tres - Desarrollo de la tarea 1.

#### Paso 4. Desarrollo de la tarea 2.

Esta tarea consiste en llenar el espacio del recuadro del Middleware controlador de Interacción en la hoja de resolución, para ello debe realizar lo siguiente:

1. Escriba la hora en la que usted dará inicio con la tarea (Figura 5a).
2. En la tarea 1, usted definió que las formas de interacción disponibles serán gestual y vocal. Ahora, usted debe seleccionar los controladores que manejarán esas interacciones, por ejemplo, para controlar los comandos gestuales es necesario el Controlador gestual, y para manejar las interacciones vocales es necesario el Controlador vocal (Figura 5b).
3. Usted debe dibujar un rectángulo para cada controlador seleccionado y escribir su respectivo nombre (Figura 5c).
4. Ahora, es necesario identificar los dispositivos a los cuales van dirigidos los comandos del usuario (foco, puerta, calefactor, radio), para ello debe elegir el componente "Identificador de dispositivo objetivo" y dibujarlo junto a los controladores de interacción seleccionados (Figura 5d).
5. Luego, en el recuadro Middleware Controlador de Interacción usted debe dibujar las conexiones que tendrá cada uno de los componentes elegidos hasta este punto conforme indica la arquitectura AALIoT. Cada dispositivo con su respectivo controlador, y los controladores con el identificador de dispositivos (Figura 5e).
6. Finalmente, escribir la hora de finalización de la tarea (Figura 5f).

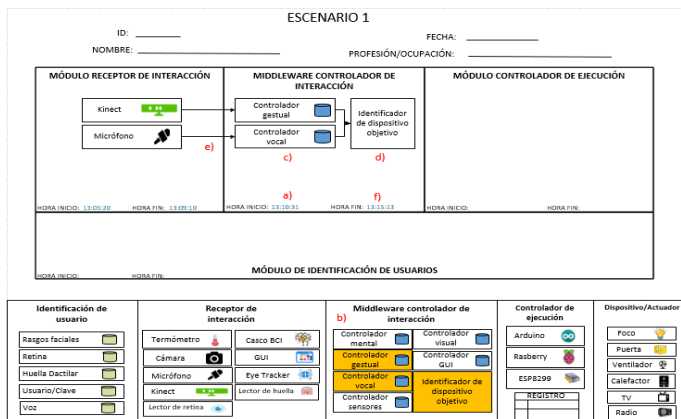


Figura 5. Paso cuatro - Desarrollo de la tarea 2.

#### Paso 5. Desarrollo de la tarea 3.

Esta tarea consiste en llenar el espacio del recuadro del Módulo Controlador de Ejecución, para ello debe realizar lo siguiente:

1. Escriba la hora en la que usted dará inicio con la tarea (Figura 6a).
2. Ahora, usted debe definir la parte electrónica de la arquitectura. Seleccione el controlador que considere adecuado, en este caso seleccionaremos el controlador "Arduino"; además, para registrar los dispositivos debe seleccionar el componente "Registro" (Figura 6b).
3. Dibujar un rectángulo para cada controlador seleccionado y escribir su respectivo nombre y, debajo, la tabla de registro (Figura 6c).
4. Ahora, es necesario seleccionar el dispositivo sobre el cual tendrá efecto el comando del usuario. Seleccionamos el dispositivo "Radio" (Figura 6d).
5. En el Módulo Controlador de Ejecución, dibujar un rectángulo y escribir el nombre del componente seleccionado (Figura 6e).
6. Ahora, debe escribir el nombre del dispositivo elegido en la tabla de registro de dispositivos y asignarle un ID (Figura 6f).
7. Luego, usted debe dibujar las conexiones que tendrá cada uno de los componentes elegidos hasta este punto conforme indica la arquitectura AALIoT (Figura 6g).
8. Finalmente, escribir la hora de finalización de la tarea (Figura 6h).

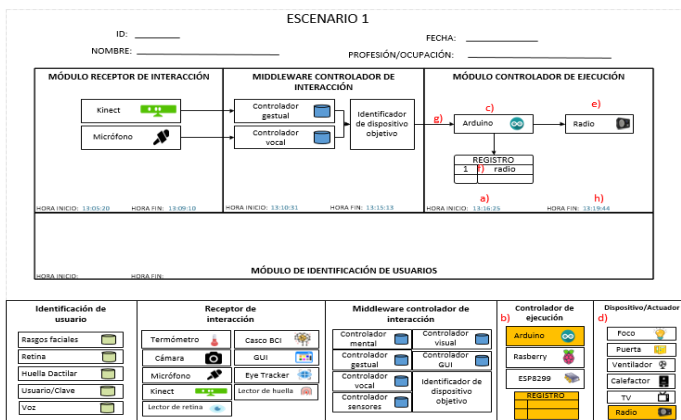


Figura 6. Paso cinco - Desarrollo de la tarea 3.

#### Paso 6. Desarrollo de la tarea 4.

Esta tarea consiste en llenar el espacio del recuadro del Módulo de Identificación de Usuarios en la hoja de resolución, para ello debe realizar lo siguiente:

1. Escriba la hora en la que usted dará inicio con la tarea (Figura 7a).
2. Ahora usted debe determinar si es necesario incluir componentes para el Módulo Identificador de Usuarios, debe hacerlo en base al ejercicio que se encuentra resolviendo, en el ejemplo es necesario que el sistema sea capaz de identificar al sujeto que está emitiendo la orden. Solamente si considera necesaria la inclusión de un módulo para la identificación de usuarios deberá continuar con los siguientes pasos.
3. Elija la forma en la que se llevará a cabo la identificación, en este ejemplo elegiremos una identificación en base a los rasgos faciales del usuario (Figura 7b).
4. En el recuadro en blanco del "Módulo de identificación de usuarios", usted debe dibujar un rectángulo y escribir el nombre del tipo de identificación elegida (Figura 7c).
5. Para poder realizar la identificación del usuario, es necesario que éste interactúe con un dispositivo que recoja las señales necesarias para dicha identificación. En este ejemplo, se captarán las imágenes del rostro del usuario con una cámara (Figura 7d).
6. Ahora, en el recuadro del Módulo Receptor de Interacción, usted debe dibujar un rectángulo para el elemento seleccionado y escribir su respectivo nombre (Figura 7e).
7. Luego, usted debe dibujar las conexiones que tendrá cada uno de los componentes elegidos hasta este punto conforme indica la arquitectura AALIoT (Figura 7f).
8. Finalmente, escribir la hora de finalización de la tarea (Figura 7g).

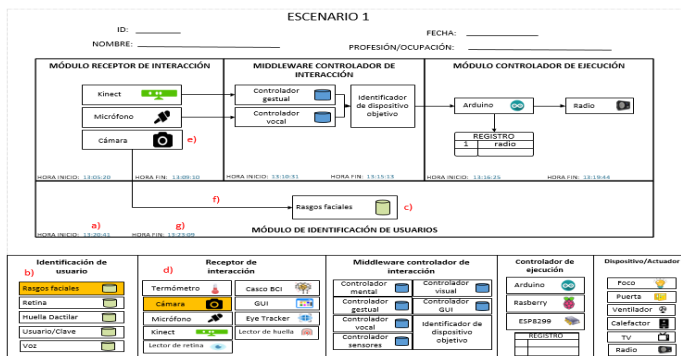


Figura 7. Paso seis - Desarrollo de la tarea 4.

### 3 Ejercicios propuestos

A continuación, se presentan los ejercicios propuestos para cada escenario planteado:

#### Ejercicio 1

José, que padece de Esclerosis Lateral Amiotrófica (ELA, la enfermedad de Stephen Hawking), se encuentra sentado en su silla de ruedas en la sala de su hogar. Su cuidador no está presente, ha anochecido, la habitación está oscura y José desea encender la bombilla de luz. Dar solución al problema de José usando los componentes de la arquitectura que considere adecuados para construir la aplicación IoT apropiada considerando las necesidades de interacción del individuo.

#### Ejercicio 2

Un paciente se encuentra en su habitación privada dentro de un hospital. Se requiere una aplicación IoT que regule la temperatura de la habitación de forma automática mientras el paciente duerme, en base a lecturas de su temperatura corporal y sus preferencias personales (caliente, frío, etc.). Usar los componentes de la arquitectura que considere adecuados para construir la aplicación IoT apropiada considerando las necesidades de interacción del individuo.

#### Ejercicio 3

Elizabeth sufre de Parkinson, por lo cual no puede coordinar sus movimientos motores. Ella desea abrir la puerta de su habitación para poder entrar; una vez dentro, desea prender la TV y sintonizar su canal favorito. Por razones de privacidad, Elizabeth prefiere ser la única persona con acceso a esa habitación; además, le gusta que la temperatura del lugar permanezca siempre a 18° C. Usar los componentes de la arquitectura que considere adecuados para construir la aplicación IoT apropiada considerando las necesidades de interacción del individuo.



## Anexo I. Descripción de la arquitectura de software AALIoT

La arquitectura está diseñada para ser el soporte de aplicaciones basadas en internet de las cosas (IoT, por sus siglas en inglés *Internet of Things*) que se despliegan en entornos de vida asistida por el entrono (AAL, por sus siglas en inglés *Ambient Assited Living*), por ejemplo, hogar, geriátricos, hospitales, etc. Estas aplicaciones usan diferentes tipos de dispositivos o sensores conectados a internet para brindar asistencia a grupos de usuarios vulnerables: adultos mayores y personas con diversidad funcional.

La arquitectura AALIoT considera los tipos de interacción natural más frecuentes en ambientes de vida asistidos: i) interacción vocal, ii) interacción visual, iii) interacción gestual, iv) interacción mental, vi) interacción basada en sensores, vii) interacción con interfaces gráficas de usuario y los integra con los diferentes dispositivos, sensores y actuadores que forman parte de las aplicaciones encargadas de brindar una mejor calidad de vida a sus usuarios. A continuación, se brinda una breve descripción de cada uno de los componentes presentados en la arquitectura de software:

### Módulo de identificación de usuarios

Se encarga de identificar al usuario dentro del entorno y habilitar las acciones que le son permitidas. Este módulo debe ser incluido solamente cuando los requerimientos de la aplicación IoT lo consideren necesario. Está constituido por los siguientes componentes:



**Identificador de usuario:** componente encargado de identificar o reconocer al usuario dentro del entorno.



**Repositorio de información de usuario:** repositorio que almacena la información personal de un usuario y datos necesarios para validar la autenticación.

### Módulo receptor de interacción

Se encarga de reconocer los comandos de los usuarios mediante dispositivos receptores de interacción o sensores para su posterior ejecución. Está constituido por los siguientes componentes:



**Dispositivo receptor de interacción:** Son todos los dispositivos capaces de captar gestos, señales auditivas, movimiento de los ojos, comandos mentales, acciones realizadas en dispositivos con GUIs, o valores medidos por sensores. Entre los dispositivos receptores de interacción, podemos encontrar los siguientes: kinect, micrófono, videocámara, casco receptor de señales EEG, sensor de temperatura, entre otros.



- **Registro de dispositivos receptores de interacción:** Se refiere a la especificación y detalle de las propiedades de cada uno de los dispositivos receptores de interacción disponibles en el entorno.



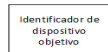
#### Middleware de interacción

Se encarga de interpretar las señales enviadas por el Módulo receptor de interacción, identificar el dispositivo o dispositivos a los cuales van dirigidas y enviar la solicitud al Controlador de ejecución. Está constituido por los siguientes componentes:



**Controlador de interacción:** Procesa e interpreta los comandos recibidos en base a algoritmos específicos y a repositorios de comandos disponibles para cada forma de interacción. Se divide en los siguientes sub-componentes:

- Controladores de interacción vocal, visual, gestual, mental, basada en sensores y GUI.
- Repositorio de comandos de voz, visuales, gestuales, mentales y GUIs.
- Repositorio de información de salud.



**Identificador de dispositivo objetivo:** Componente que identifica el dispositivo o actuador al que va dirigido el comando enviado por el usuario.

#### Módulo controlador de ejecución

Aquí se lleva a cabo la ejecución de los comandos enviados por el usuario. Cada dispositivo o actuador cambiará de estado según la solicitud del usuario. Está constituido por los siguientes componentes:



**Controlador de ejecución:** Prepara y envía la acción a ejecutar al dispositivo o actuador objetivo. Por ejemplo, un controlador Arduino o un RaspberryPi.

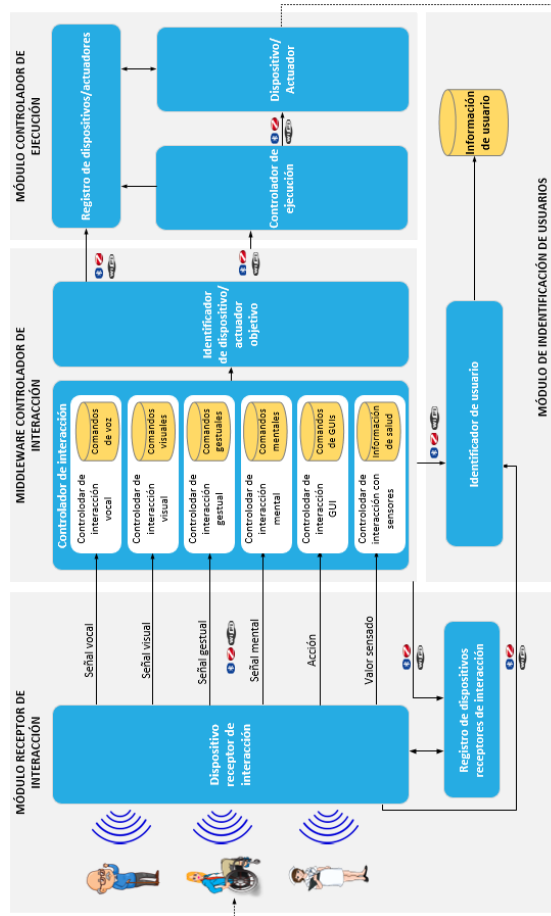


**Registro de dispositivo objetivo:** Se refiere a la especificación y detalle de las propiedades y comandos que aceptan cada uno de los dispositivos o actuadores presentes en el entorno.



**Dispositivo objetivo:** Representa a los diferentes dispositivos que pueden ser controlados dentro del entorno, por ejemplo: bombilla de luz, puerta eléctrica, persiana eléctrica, ventana eléctrica, ventilador, etc.

## Anexo II. Arquitectura de Software





### Anexo III. Guía del procedimiento de evaluación

#### Paso 1. Datos generales.

ESCENARIO 1	
ID: _____	FECHA: _____
NOMBRE: _____	PROFESIÓN/OCUPACIÓN: _____

Paso 1. Datos generales en el encabezado de la hoja de resolución.













#### Paso 2. Lectura del ejercicio propuesto.

##### Ejercicio ejemplo

Alex es una persona con discapacidad física: no posee sus dos piernas. Alex desea encender su radio personal y sintonizar una emisora sin necesidad de acercarse a ella. La radio se encenderá únicamente cuando sea Alex el que dé la orden haciendo caso omiso a las solicitudes de otras personas. Usar los componentes de la arquitectura que considere adecuados para construir la aplicación IoT apropiada considerando las necesidades de interacción del individuo.

Paso 2. Ejercicio ejemplo basado en un escenario específico.

#### Paso 3. Desarrollo de la tarea 1.

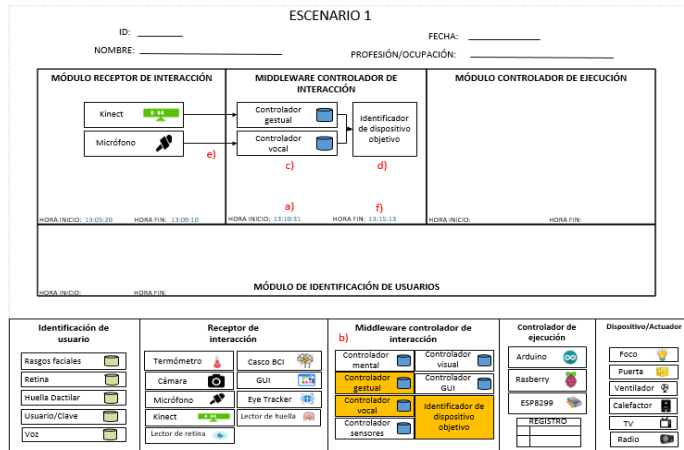
ESCENARIO 1				
ID: _____		FECHA: _____		
NOMBRE: _____		PROFESIÓN/OCUPACIÓN: _____		
<b>MÓDULO RECEPTOR DE INTERACCIÓN</b>  c)    h)  i)  j)  k)  l)   Módulo de identificación de usuarios	<b>MIDDLEWARE CONTROLADOR DE INTERACCIÓN</b>  b)  c)  d)  e)  f)   Middleware controlador de interacción	<b>MÓDULO CONTROLADOR DE EJECUCIÓN</b>  Controlador mental Controlador gestual Controlador vocal Controlador sensores  Controlador visual Controlador GUI Identificador de dispositivo objetivo  Controlador de ejecución	<b>Dispositivo/Actuador</b>  Arduino Raspberry ESP8266 REGISTRO  Foco Puerta Ventilador Calentador TV Radio	

Paso 3. Desarrollo de la tarea 1.



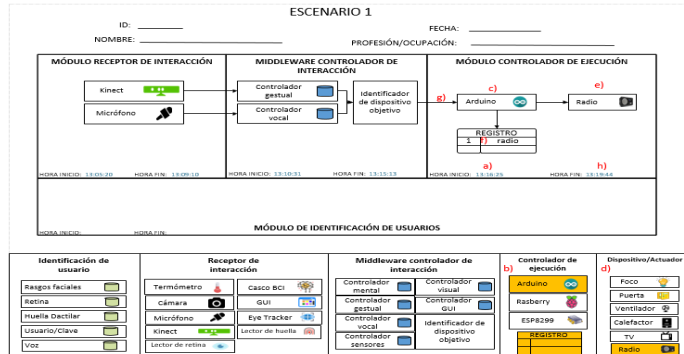


Paso 4. Desarrollo de la tarea 2.



Paso 4. Desarrollo de la tarea 2.

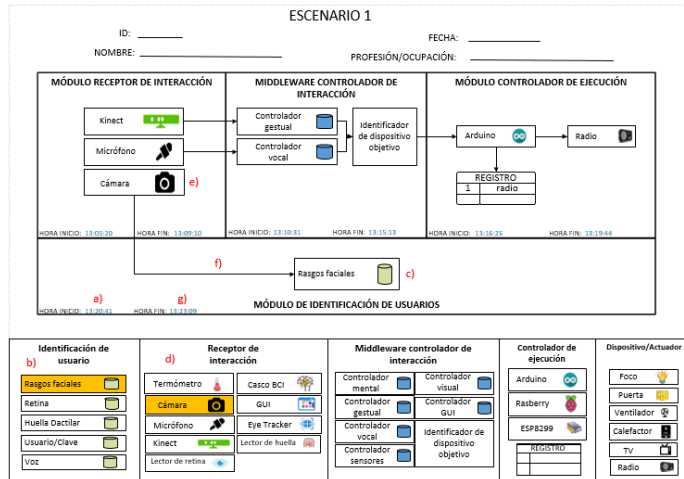
Paso 5. Desarrollo de la tarea 3.



Paso 5. Desarrollo de la tarea 3.



Paso 6. Desarrollo de la tarea 4.



Paso 6. Desarrollo de la tarea 4.





## E.2. Cuestionario

Evaluación de Arquitectura enfocada en la Interacción Hombre-Maquina e Internet de las Cosas(IoT) en Ambientes de Vida Asistidos(AAL)

16/8/18 11:58 p.m.

### Evaluación de Arquitectura enfocada en la Interacción Hombre-Maquina(HCI) para aplicaciones de Internet de las Cosas(IoT) en Ambientes de Vida Asistidos(AAL)

Para cada una de las preguntas marque, por favor, una cruz sobre el círculo que se encuentra lo más cerca posible de su opinión.

LEA POR FAVOR CADA PREGUNTA CUIDADOSAMENTE ANTES DE DAR SU RESPUESTA

**\*Obligatorio**

1. **1. La forma de crear la arquitectura me ha parecido compleja y difícil de seguir \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente de Acuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente en Desacuerdo

2. **2. Creo que la arquitectura reduciría el tiempo y el esfuerzo requerido para crear sistemas enfocados en HCI para aplicaciones de IoT en AAL \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo

3. **3. De manera general, la arquitectura(módulos y componentes) es difícil de entender \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente de Acuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente en Desacuerdo

4. **4. Los pasos a seguir para crear la arquitectura son claros y fáciles de entender \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo



5. **5. De manera general, considero que la arquitectura es útil \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo

6. **6. La arquitectura es difícil de aprender \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Totalmente de Acuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente en Desacuerdo

7. **7. Creo que la forma en que los componentes interactúan entre sí en la arquitectura es útil para la creación de sistemas enfocados en HCI para aplicaciones de IoT en AAL \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo

8. **8. Si tuviera que utilizar una arquitectura enfocada en HCI para aplicaciones de IoT en AAL en el futuro, creo que tendría en cuenta esta arquitectura \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo

9. **9. Creo que la arquitectura NO es lo suficientemente expresiva para definir como los componentes interactúan entre sí \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Totalmente de Acuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente en Desacuerdo

10. **10. El uso de esta arquitectura mejoraría mi rendimiento en la creación de sistemas enfocados en HCI para aplicaciones de IoT en AAL \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo



11. **Pienso que sería fácil ser hábil usando esta arquitectura \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo

12. **De manera general, pienso que esta arquitectura NO puede cubrir adecuadamente los requisitos para sistemas enfocados en HCI para IoT en AAL \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente de Acuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente en Desacuerdo

13. **En caso de tener la necesidad de crear sistemas enfocados en HCI para aplicaciones de IoT en AAL, tendría la intención de utilizar esta arquitectura en el futuro \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en Desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de Acuerdo

14. **No recomendaría el uso de esta arquitectura \***

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente de Acuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente en Desacuerdo

15. **¿Tiene alguna sugerencia de cómo hacer que esta arquitectura sea más fácil de usar? \***

---



---



---



---



---



16. **16. ¿Cuáles son las razones por las que tiene o no la intención de usar esta arquitectura en un futuro? \***

---

---

---

---

---

17. **Introduzca por favor su nombre \***

---





### E.3. Resultados primer cuasi-experimento

ANÁLISIS DE EFECTIVIDAD					
ID	NOMBRES	EJ1	EJ2	EJ3	TOTAL
1	Bryan Coronel	0,875	1	0,875	0,92
2	David Morocho	1	0,75	0,75	0,83
3	David Piedra	1	1	1	1,00
4	Edwin Cabrera	1	1	0,75	0,92
5	Elio Quevedo	1	0,75	0,875	0,88
6	Elizabeth Sari	0,75	0,875	0,5	0,71
7	Esteban Espinoza	1	0,875	0,75	0,88
8	Jorge Vázquez	1	0,75	0,875	0,88
9	Paola Cárdenas	1	0,5	1	0,83
10	Pedro Tenezaca	0,75	1	0,625	0,79
11	Steven Godoy	1	1	1	1,00
12	Yolanda Aucapiña	0,875	0,625	0,625	0,71

Tabla E.1: Resultados efectividad de egresados/profesionales

ANÁLISIS DE EFICIENCIA					
ID	NOMBRES	EJ1	EJ2	EJ3	TOTAL
1	Bryan Coronel	4,00	4,00	7,00	15,00
2	David Morocho	1,87	2,72	5,62	10,20
3	David Piedra	1,97	2,98	4,23	9,18
4	Edwin Cabrera	3,77	4,05	4,50	12,32
5	Elio Quevedo	2,08	2,13	4,03	8,25
6	Elizabeth Sari	5,00	4,00	6,00	15,00
7	Esteban Espinoza	3,30	4,38	4,88	12,57
8	Jorge Vázquez	3,00	4,00	6,00	13,00
9	Paola Cárdenas	3,27	3,92	4,67	11,85
10	Pedro Tenezaca	4,00	5,50	4,50	14,00
11	Steven Godoy	4,00	4,00	5,00	13,00
12	Yolanda Aucapiña	3,00	2,78	5,60	11,38

Tabla E.2: Resultados eficiencia de egresados/profesionales





## E.4. Resultados segundo cuasi-experimento

ANÁLISIS DE EFECTIVIDAD					
ID	NOMBRES	EJ1	EJ2	EJ3	TOTAL
1	Belen Toledo	0,75	0,5	0,75	0,67
2	Bryan Aguilar	1	0,875	0,875	0,92
3	Bryan Mora	1	0,75	1	0,92
4	Carlos Valladarez	0,875	0,625	0,625	0,71
5	Christian Salinas	0,625	0,875	0,75	0,75
6	Christian Torres	0,625	0,625	0,75	0,67
7	David Buñay	1	0,75	0,625	0,79
8	David Valladares	0,625	0,375	0,375	0,46
9	Edisson Sigua	0,625	0,75	0,875	0,75
10	Emily Arteaga	0,375	0,625	0,75	0,58
11	Freddy Abad	0,75	0,5	0,75	0,67
12	Gabriela Delgado	1	1	0,875	0,96
13	Gustavo Mora	0,25	0,375	1	0,54
14	Jefferson Arias	0,375	0,5	0,875	0,58
15	Luis Muñoz	0,875	0,5	0,75	0,71
16	Mateo Quizhpi	0,625	0,625	0,5	0,58
17	Paola Remache	0,75	0,5	0,625	0,63
18	Tania Landivar	0,875	0,75	1	0,88

Tabla E.3: Resultados efectividad de estudiantes

ANÁLISIS DE EFICIENCIA					
ID	NOMBRES	EJ1	EJ2	EJ3	TOTAL
1	Belen Toledo	3,35	4,57	5,87	13,78
2	Bryan Aguilar	4,00	4,00	7,50	15,50
3	Bryan Mora	7,00	3,00	8,00	18,00
4	Carlos Valladarez	4,50	3,00	3,00	10,50
5	Christian Salinas	8,00	10,00	7,00	25,00
6	Christian Torres	9,00	7,00	6,50	22,50
7	David Buñay	7,00	4,00	7,00	18,00
8	David Valladarez	9,00	6,00	6,00	21,00
9	Edisson Sigua	13,00	8,00	14,00	35,00
10	Emily Arteaga	5,00	6,00	5,00	16,00
11	Freddy Abad	2,72	1,17	4,00	7,88



12	Gabriela Delgado	2,50	3,00	6,50	12,00
13	Gustavo Mora	2,23	3,38	3,52	9,13
14	Jefferson Arias	12,50	1,50	6,50	20,50
15	Luis Muñoz	6,00	4,00	7,00	17,00
16	Mateo Quizhpi	6,00	4,00	4,50	14,50
17	Paola Remache	4,00	5,50	5,50	15,00
18	Tania Landivar	11,00	4,50	6,50	22,00

Tabla E.4: Resultados eficiencia de estudiantes



# Bibliografía

- [1] AAL-EUROPE (2016). Active and assisted living programme. <http://www.aal-europe.eu/about/objectives/>.
- [2] Abascal, J. (2002). Human-computer interaction in assistive technology: from "patchwork" to universal design". In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, volume 3, pages 6–pp. IEEE.
- [3] Abrahão, S., Insfran, E., Carsí, J. A., and Genero, M. (2011). Evaluating requirements modeling methods based on user perceptions: A family of experiments. *Information Sciences*, 181(16):3356–3378.
- [4] Ali Babar, M. (2008). Assessment of a framework for designing and evaluating security sensitive architecture.
- [5] Anthony, M. (2018). Voice interfaces: New era of human-computer-interactions. <https://bit.ly/2y40lfi>.
- [6] Ashton, K. et al. (2009). That ‘internet of things’ thing. *RFID journal*, 22(7):97–114.
- [7] Babar, M. A. and Kitchenham, B. (2007). The impact of group size on software architecture evaluation: a controlled experiment. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, pages 420–429. IEEE.
- [8] Babar, M. A., Kitchenham, B., and Jeffery, R. (2008). Comparing distributed and face-to-face meetings for software architecture evaluation: A controlled experiment. *Empirical Software Engineering*, 13(1):39–62.
- [9] Babar, M. A., Winkler, D., and Biffi, S. (2007). Evaluating the usefulness and ease of use of a groupware tool for the software architecture evaluation process. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, pages 430–439. IEEE.



- [10] Basili, V., Caldiera, G., and Rombach, H. D. (1994). Encyclopedia of software engineering.
- [11] Basili, V. R. (1993). The experimental paradigm in software engineering. In *Experimental Software Engineering Issues: Critical Assessment and Future Directions*, pages 1–12. Springer.
- [12] Basili, V. R., Selby, R. W., and Hutchens, D. H. (1986). Experimentation in software engineering. *IEEE Transactions on software engineering*, (7):733–743.
- [13] Bass, L., Clements, P., and Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.
- [14] Bengtsson, PerOlof y Lassing, N. y. B. J. y. v. V. H. Análisis de modificabilidad de nivel de arquitectura (alma). *Diario de Sistemas y Software*, (1-2).
- [15] Botia, J. A., Villa, A., and Palma, J. (2012). Ambient assisted living system for in-home monitoring of healthy independent elders. *Expert Systems with Applications*, 39(9):8136–8148.
- [16] Bryant, D. P. and Bryant, B. R. (2003). *Assistive technology for people with disabilities*. Allyn and Bacon Boston.
- [17] Cantoni, V., Cellario, M., and Porta, M. (2004). Perspectives and challenges in e-learning: towards natural interaction paradigms. *Journal of Visual Languages & Computing*, 15(5):333–345.
- [18] Carroll, J. M. (2013). Human computer interaction-brief intro. *The Encyclopedia of Human-Computer Interaction*, 2nd Ed.
- [19] Cedillo, P. (2016). Monitorización de calidad de servicios cloud mediante modelos en tiempo de ejecución.
- [20] Choi, J., Park, J. S., Chang, S., and Lee, H. R. (2017). Multi-purpose connected electronic nose system for health screening and indoor air quality monitoring. In *Information Networking (ICOIN), 2017 International Conference on*, pages 495–499. IEEE.
- [21] Colantonio, S., Coppini, G., Giorgi, D., Morales, M.-A., and Pascali, M. A. (2018). Computer vision for ambient assisted living: Monitoring systems for personalized healthcare and wellness that are robust in the real world and accepted by users, carers, and society. In *Computer Vision for Assistive Healthcare*, pages 147–182. Elsevier.



- [22] CONADIS (2018). Personas con discapacidad registradas. <https://bit.ly/2IHdtIo>.
- [23] Cook, T. D. and Campbell, D. T. (1979). *Quasi-experimentation: Design and analysis for field settings*, volume 3. Rand McNally Chicago.
- [24] Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. PhD thesis, Massachusetts Institute of Technology.
- [25] Dix, A. (2009). Human-computer interaction. In *Encyclopedia of database systems*, pages 1327–1331. Springer.
- [26] Dohr, A., Modre-Opsrian, R., Drobics, M., Hayn, D., and Schreier, G. (2010). The internet of things for ambient assisted living. In *Information technology: new generations (ITNG), 2010 seventh international conference on*, pages 804–809. Ieee.
- [27] Du, H., Henry, P., Ren, X., Cheng, M., Goldman, D. B., Seitz, S. M., and Fox, D. (2011). Interactive 3d modeling of indoor environments with a consumer depth camera. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 75–84. ACM.
- [28] Errobidart, J., Uriz, A. J., Gonzalez, E., Gelosi, I. E., and Etcheverry, J. A. (2017). Offline domotic system using voice comands. In *Embedded Systems (CASE), 2017 Eight Argentine Symposium and Conference on*, pages 1–6. IEEE.
- [29] Fischer, G. (2001). User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1-2):65–86.
- [30] Fishbein, M. and Ajzen, I. (1975). *Belief, attitude, intention and behavior: An introduction to theory and research*.
- [31] Flórez Revuelta, F. (2008). Vida asistida por el entorno. *Revista Informativa de la Asociación Profesional Española de Terapeutas Ocupacionales*, pages 12–17.
- [32] Gorschek, T., Garre, P., Larsson, S., and Wohlin, C. (2006a). A model for technology transfer in practice. *IEEE software*, 23(6):88–95.
- [33] Gorschek, T., Garre, P., Larsson, S., and Wohlin, C. (2006b). A model for technology transfer in practice. *IEEE software*, 23(6):88–95.



- [34] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.
- [35] Group, O. M. (2003 (accessed August, 2018)). Omg unified modeling language specification. [www.rational.com/uml/resources/documentation/](http://www.rational.com/uml/resources/documentation/).
- [36] Grudin, J. (1992). Utility and usability: research issues and development contexts. *Interacting with computers*, 4(2):209–217.
- [37] Hohmann, L. (2003). *Beyond software architecture: creating and sustaining winning solutions*. Addison-Wesley Longman Publishing Co., Inc.
- [38] iMotion (2010 (accessed December 7, 2014)). Eyetracking monitoring. <https://bit.ly/2sPhLYf>.
- [39] INEC (2016). La población adulto mayor en la ciudad de quito: Estudio de la situación sociodemográfica y socioeconómica.
- [40] ITU (2012 (accessed Junio 15, 2018)). Descripción general de internet de los objetos. <https://www.itu.int/rec/T-REC-Y.2060-201206-I>.
- [41] Iturrate, I., Escolano, C., Antelis, J., and Minguez, J. (2009). Dispositivos robóticos de rehabilitación basados en interfaces cerebro-ordenador: silla de ruedas y robot para teleoperación. In *III International Congress on Domotics, Robotics and Remote-Assistance for All, Barcelona, Spain*, pages 124–134.
- [42] Jagadish, B., Kiran, M., and Rajalakshmi, P. (2017). A novel system architecture for brain controlled iot enabled environments. In *e-Health Networking, Applications and Services (Healthcom), 2017 IEEE 19th International Conference on*, pages 1–5. IEEE.
- [43] Jara, A. J., Zamora, M. A., and Skarmeta, A. F. (2011). An internet of things-based personal device for diabetes therapy management in ambient assisted living (aal). *Personal and Ubiquitous Computing*, 15(4):431–440.
- [44] Karray, F., Alemzadeh, M., Saleh, J. A., and Arab, M. N. (2008). Human-computer interaction: Overview on state of the art.
- [45] Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele Universit.



- [46] Ko, Y.-J., Huang, H.-M., Hsing, W.-H., Chou, J., Chiu, H.-C., and Ma, H.-P. (2015). A patient-centered medical environment with wearable sensors and cloud monitoring. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 628–633. IEEE.
- [47] Kuneva, M., Dózsa, C., Mann, E., Miles, I., Parent, A., and Vasconcelos, D. (2010). Interim evaluation of the ambient assisted living joint programme: Unlocking innovation in ageing well. *Report of High-Level Expert Panel, appointed by the European Commission/DG Information Society & Media*, 77.
- [48] Leap, M. (2015). Leap motion controller. <http://www.leapmotion.com>.
- [49] Mainetti, L., Manco, L., Patrono, L., Secco, A., Sergi, I., and Vergallo, R. (2016). An ambient assisted living system for elderly assistance applications. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*, pages 1–6. IEEE.
- [50] Mead, N. R., Ellison, R. J., Linger, R. C., Longstaff, T., and McHugh, J. (2000). Survivable network analysis method. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
- [51] Microsoft (2011). Kinect camera. <http://www.xbox.com/en-US/>.
- [52] MIES (2016). Dirección población adulta mayor. <https://bit.ly/2A0f7IF>.
- [53] Mitchell, B. S. and Mancoridis, S. (2006). On the automatic modularization of software systems using the bunch tool. *IEEE Transactions on Software Engineering*, 32(3):193–208.
- [54] Mitchell, B. S. and Mancoridis, S. (2008). On the evaluation of the bunch search-based software modularization algorithm. *Soft Computing*, 12(1):77–93.
- [55] Mitchell, J. (1999). *From telehealth to e-health: the unstoppable rise of e-health*. Department of Communications, Information Technology and the Arts.
- [56] Moody, D. (2001). *A Practical Method for Representing Large Entity Relationship Models*. PhD thesis, PhD Thesis.
- [57] MYO. Gesture control armband — wearable technology. <https://www.myo.com/>.



- [58] Nations, U. (2015). World population prospects: The 2015 revision. *United Nations Econ Soc Aff*, 33(2):1–66.
- [59] Oliveira, T. J. M., Costa, Â., Neves, J., and Novais, P. (2013). A comprehensive clinical guideline model and a reasoning mechanism for aal systems. *International journal of artificial intelligence*, 11(A13):57–73.
- [60] ONU (2017). Envejecimiento. <https://bit.ly/2t63EfD>.
- [61] Parra Ullaauri, J. M. and Valdez Solís, W. F. (2017). Diseño e implementación de un prototipo de pastillero digital para la toma de medicación de pacientes adultos mayores, mediante la integración de nuevas tecnologías de hardware y software libre. B.S. thesis.
- [62] Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.
- [63] Philip, M. (2017 (accessed August, 2018)). Eye tracking software: How science helps you see through customer’s eyes. <https://bit.ly/2LFC5Co>.
- [64] Pires, C. G., Pinto, F., Teixeira, V. D., Freitas, J., and Dias, M. S. (2012). Living home center—a personal assistant with multimodal interaction for elderly and mobility impaired e-inclusion. *Proc. PROPOR. Coimbra*.
- [65] Rahman, M. A. and Hossain, M. S. (2018). A cloud-based virtual caregiver for elderly people in a cyber physical iot system. *Cluster Computing*, pages 1–14.
- [66] Ramón, C., Antonio, J., Torres, F., and Candelas-Herías, F. A. (2006). Tecnologías en la inteligencia ambiental.
- [67] Rayes, A. and Salam, S. (2017). The things in iot: Sensors and actuators. In *Internet of Things From Hype to Reality*, pages 57–77. Springer.
- [68] Reeves, B. and Nass, C. I. (1996). *The media equation: How people treat computers, television, and new media like real people and places*. Cambridge university press.
- [69] Rescher, N. (1973). The primacy of practice. Basil Blackwell.
- [70] Riva, G., Loret, P., Lunghi, M., Vatalaro, F., and Davide, F. (2003). Presence 2010: The emergence of ambient intelligence. *Emerging Communication*, 5:59–84.





- [71] Rodríguez-Porrero, C. (2003). Discapacidad y calidad de vida. *Rev Mult Gerontol*, 10(2):66–9.
- [72] Rozo, C. (2010). Consideraciones éticas de la tecnología de asistencia en personas en condición de discapacidad: Posibilitar o limitar la autonomía? *Revista Latinoamericana de Bioética*, 10(18):056–065.
- [73] Ruiz, I., Castillo, V. H., Gutiérrez, J. R., Villanueva, M., and Álvarez, J. L. (2014). Estructuras de representación del conocimiento en ambientes asistidos. *Research in Computing Science*, 76:73–78.
- [74] Schmidt, A. (2000). Implicit human computer interaction through context. *Personal technologies*, 4(2-3):191–199.
- [75] Skillen, K.-L., Chen, L., Nugent, C. D., Donnelly, M. P., Burns, W., and Solheim, I. (2012). Ontological user profile modeling for context-aware application personalization. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 261–268. Springer.
- [76] Soegaard, M. and Dam, R. F. (2012). The encyclopedia of human-computer interaction. *The Encyclopedia of Human-Computer Interaction*.
- [77] Steinke, F., Fritsch, T., Brem, D., and Simonsen, S. (2012). Requirement of aal systems: older persons’ trust in sensors and characteristics of aal technologies. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*, page 15. ACM.
- [78] Tomii, S. and Ohtsuki, T. (2012). Falling detection using multiple doppler sensors. In *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, pages 196–201. IEEE.
- [79] Valera, A. J. J., Zamora, M. A., and Skarmeta, A. F. (2010). An architecture based on internet of things to support mobility and security in medical environments. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE.
- [80] Valli, A. (2008). The design of natural interaction. *Multimedia Tools and Applications*, 38(3):295–305.
- [81] Watanabe, H., Kawarasaki, M., Sato, A., and Yoshida, K. (2012). Development of wearable heart disease monitoring and alerting system associated with smartphone. In *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, pages 292–297. IEEE.



- [82] WHO (2011). World report on disability. *Geneva: WHO*.
- [83] Williams, L. G. and Smith, C. U. (2002). Pasa sm: a method for the performance assessment of software architectures. In *Proceedings of the 3rd international workshop on Software and performance*, pages 179–189. ACM.
- [84] Wohlin, C. (2007). Introduction to aggregation of case studies why aggregation. Sweden.
- [85] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002a). Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791.
- [86] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002b). Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791.
- [87] Woods, M. D., Kirk, M. D., Agarwal, M. S., Annandale, E., Arthur, T., Harvey, J., Hsu, R., Katbamna, S., Olsen, R., Smith, L., et al. (2005). Vulnerable groups and access to health care: a critical interpretive review. *National Coordinating Centre NHS Service Delivery Organ RD (NCCSDO) Retrieved May, 27:2012*.
- [88] Wu, Y. and Huang, T. S. (1999). Vision-based gesture recognition: A review. In *International Gesture Workshop*, pages 103–115. Springer.
- [89] Yang, C.-T., Liu, J.-C., Liao, C.-J., Wu, C.-C., and Le, F.-Y. (2013a). On construction of an intelligent environmental monitoring system for health-care. In *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2013 International Conference on*, pages 246–253. IEEE.
- [90] Yang, J., Xiong, N., and Vasilakos, A. V. (2013b). Two-stage enhancement scheme for low-quality fingerprint images by learning from the images. *IEEE transactions on human-machine systems*, 43(2):235–248.
- [91] Zukerman, I. and Albrecht, D. W. (2001). Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):5–18.